

UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL  
CUSCO  
FACULTAD DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA, INFORMÁTICA  
Y MECÁNICA  
ESCUELA PROFESIONAL DE INGENIERÍA ELÉCTRICA



T E S I S

---

**PROGRAMACIÓN PARAMÉTRICA–VISUAL APLICADA  
A SOFTWARE BIM PARA EL DESARROLLO DE LA  
ETAPA DE DISEÑO Y CÁLCULO DE INSTALACIONES  
ELÉCTRICAS INTERIORES**

---

**PRESENTADO POR:**

Br. Luis Giuseppe Yucra Altamirano

**Para optar al Título Profesional de  
INGENIERO ELECTRICISTA**

**ASESORA:**

Mgt. Ing. Mary Elisa Barrionuevo Prado

CUSCO – PERÚ  
2023

# INFORME DE ORIGINALIDAD

(Aprobado por Resolución Nro. CU-303-2020-UNSAAC)

El que suscribe, Asesor del trabajo de investigación/tesis titulada: PROGRAMACIÓN  
PARAMÉTRICA-VISUAL APLICADA A SOFTWARE BIM PARA EL DESARROLLO DE  
LA ETAPA DE DISEÑO Y CÁLCULO DE INSTALACIONES ELÉCTRICAS INTERIORES  
presentado por: LUIS GIUSEPPE YUCA ACDAMIANO con DNI Nro.: 74311506  
presentado por: --- con DNI Nro.: ---  
para optar el título profesional/grado académico de INGENIERO  
ELECTRICISTA

Informo que el trabajo de investigación ha sido sometido a revisión por 0/ veces, mediante el Software Antiplagio, conforme al Art. 6° del **Reglamento para Uso de Sistema Antiplagio de la UNSAAC** y de la evaluación de originalidad se tiene un porcentaje de 0 %.

Evaluación y acciones del reporte de coincidencia para trabajos de investigación conducentes a grado académico o título profesional, tesis

Porcentaje	Evaluación y Acciones	Marque con una (X)
Del 1 al 10%	No se considera plagio.	X
Del 11 al 30 %	Devolver al usuario para las correcciones.	
Mayor a 31%	El responsable de la revisión del documento emite un informe al inmediato jerárquico, quien a su vez eleva el informe a la autoridad académica para que tome las acciones correspondientes. Sin perjuicio de las sanciones administrativas que correspondan de acuerdo a Ley.	

Por tanto, en mi condición de asesor, firmo el presente informe en señal de conformidad y adjunto la primera página del reporte del Sistema Antiplagio.

Cusco, 13 de DICIEMBRE de 2023



Firma

Post firma INGE. MARY ELISA BAEZ NOVELO PRADO

Nro. de DNI 23811434

ORCID del Asesor 0000-0001-7760-9381

Se adjunta:

1. Reporte generado por el Sistema Antiplagio.
2. Enlace del Reporte Generado por el Sistema Antiplagio: OID: 27259:294527055

<https://UNSAAC.turnitin.com/viewer/submissions/oid:27259:294527055?locale=es-MX>

NOMBRE DEL TRABAJO

**TESIS\_LUIS YUCRA 12-23.pdf**

AUTOR

**Luis Giuseppe Yucra Altamirano**

RECUENTO DE PALABRAS

**57596 Words**

RECUENTO DE CARACTERES

**315081 Characters**

RECUENTO DE PÁGINAS

**300 Pages**

TAMAÑO DEL ARCHIVO

**11.7MB**

FECHA DE ENTREGA

**Dec 11, 2023 10:02 AM GMT-5**

FECHA DEL INFORME

**Dec 11, 2023 10:08 AM GMT-5****● 0% de similitud general**

Esta entrega no coincidió con ningún contenido comparado.

- 0% Base de datos de Internet
- Base de datos de Crossref
- 0% Base de datos de trabajos entregados
- 0% Base de datos de publicaciones
- Base de datos de contenido publicado de Crossref

**● Excluir del Reporte de Similitud**

- Material bibliográfico
- Material citado
- Material citado
- Coincidencia baja (menos de 20 palabras)

## RESUMEN

Esta tesis se enfoca en abordar la complejidad inherente al proceso de diseño y cálculo de instalaciones eléctricas en el contexto del Software BIM. Se ha desarrollado una herramienta informática que integra programación paramétrica-visual con el propósito de simplificar el uso del software BIM y aumentar la productividad en la industria de la construcción.

Se llevó a cabo un análisis exhaustivo del software BIM y se evaluaron las interfaces de programación paramétrica-visual disponibles en el mercado. Se desarrollaron módulos específicos diseñados para el modelado y cálculo de instalaciones eléctricas, con algoritmos adaptados a las condiciones constructivas en Perú.

Los resultados son evidentes: la exitosa integración de esta solución en una base arquitectónica real demuestra la viabilidad de agilizar la generación de modelos BIM y los cálculos correspondientes en la especialidad de instalaciones eléctricas. Esta simplificación del proceso de diseño y cálculo de instalaciones eléctricas dentro del entorno del Software BIM tiene un impacto significativo, logrando una reducción del tiempo de generación de modelos BIM a tan solo segundos, en línea con los alcances normativos peruanos y la metodología BIM.

***Palabras Claves:** Programación paramétrica-visual, Software BIM, Diseño y Cálculo de instalaciones eléctricas interiores, automatización*

## ABSTRACT

This thesis focuses on addressing the inherent complexity in the design and calculation process of electrical installations within the context of BIM software. A computer tool has been developed to integrate parametric-visual programming, aiming to simplify the use of BIM software and enhance productivity in the construction industry.

A comprehensive analysis of BIM software was conducted, and available parametric-visual programming interfaces in the market were evaluated. Specific modules were designed for the modeling and calculation of electrical installations, with algorithms tailored to the construction conditions in Peru.

The results are clear: the successful integration of this solution into a real architectural context demonstrates the feasibility of expediting the generation of BIM models and corresponding calculations in the field of electrical installations. This simplification of the design and calculation process within the BIM software environment has a significant impact, achieving a reduction in the generation time of BIM models to just seconds, in line with Peruvian regulatory requirements and the BIM methodology.

**Keywords:** *parametric-visual programming, BIM Software, Design and calculation of indoor electrical installations, automatization*

## DEDICATORIA

*“Que fuese epitafio del hombre más sabio:  
Yo solo pasé por aquí”  
– Santi Balmes*

A aquel misterioso ente que guía los hilos del destino  
por conspirar a mi favor en la realización de esta Tesis.  
A mis seres queridos que han trascendido hacia la eternidad,  
sé que sus espíritus están presentes y orgullosos de este logro.  
A mis amados padres, José y Dina,  
cuyo apoyo inquebrantable ha tejido  
la trama de esta emocionante aventura.  
A mis queridas hermanitas, Sheyla y Ximenita,  
a quienes mi amor excede las palabras.  
A todos aquellos que tímidamente me llaman “Amigo”  
Esta Tesis es un testimonio de esfuerzo, amor y  
perseverancia compartido con cada uno de ustedes.

Con especial cariño, extendiendo esta dedicatoria a Luis Yucra Altamirano,  
un compañero de travesías que se convirtió en un amigo entrañable y,  
en ocasiones, en mi más valioso desafío.  
A lo largo de nuestra historia compartida,  
he llegado a conocer sus facetas más profundas y a apreciar su singularidad,  
incluso más allá de lo que él mismo estaría dispuesto a revelar.  
Nuestra relación es un equilibrio delicado entre camaradería y rivalidad,  
una danza constante que nos ha llevado a superar obstáculos y celebrar victorias.  
A ti, Luis,  
te dedico este logro alcanzado,  
como testigo de nuestras travesías conjuntas y  
como cómplice de los desafíos superados.  
¡Lo hemos conseguido juntos!

## AGRADECIMIENTOS

El autor de la presente tesis, expresa su profundo agradecimiento a:

La **Escuela Profesional de Ingeniería Eléctrica** por la gran formación en tópicos varios dedicados a la ingeniería en instalaciones eléctricas y la programación digital; misma que influyó positivamente en el desarrollo del presente trabajo.

A la **Ing. Mary Elisa Barrionuevo Prado**; por su instrucción, dedicación e interés como asesora en el planteamiento y desarrollo de esta tesis.

A los **ingenieros Pablo Apaza Huanza y Vladimiro Canal Bravo**, quienes fueron primeros revisores del presente trabajo; sus primeras observaciones permitieron alcanzar la calidad investigativa esperada y obtener mejores resultados.

A la **Arq. Luciana Jordan Ramos**, por permitir hacer uso de una base arquitectónica de su autoría sobre la cual aplicar la propuesta desarrollada.

A la **Arq. Fiorella Tintaya Tintaya**, ya que ofreció su colaboración con enfoque arquitectónico para la revisión de apartados técnicos del funcionamiento de la propuesta.

A mi señor padre, **Jose Luis Yucra Zea**, por colaborar con la revisión de ortografía y gramática dentro de la redacción a lo largo de la presente tesis.

Y finalmente a los siguientes **desarrolladores de paquetería *Dynamo***:

- |                    |                  |                        |
|--------------------|------------------|------------------------|
| ■ dimitar.ven      | ■ Gavin Crump    | ■ Chuong Ho            |
| ■ Gytaco           | ■ erfajo         |                        |
| ■ tpoover          | ■ kstheanhdo     | ■ John Pierson         |
| ■ Elayoubi Mostafa | ■ Konrad K Sobon | ■ Alban de Chasteigner |

Su aporte a la comunidad de programación paramétrica-visual permitió desarrollar scripts totalmente funcionales para los objetivos de esta tesis.

## PRESENTACIÓN

**Señor:**

Decano de la Facultad de Ingeniería Eléctrica, Electrónica, Informática y Mecánica.

**Señores:**

Miembros del Comité Dictaminante de Tesis.

Declarando la total conformidad al Reglamento de Grados y Títulos de la Escuela Profesional de Ingeniería Eléctrica de la Universidad Nacional de San Antonio Abad del Cusco, vigente al año de presentación de este trabajo; el autor, aspirante al título de Ingeniero Electricista, pone a distinguida consideración suya la presente tesis titulada:

*Programación Paramétrica–Visual aplicada a software BIM para el desarrollo de la Etapa de Diseño y Cálculo de Instalaciones Eléctricas Interiores*

Trabajo de Tesis cuyo planteamiento se halla dentro del enfoque temático del diseño de instalaciones eléctricas interiores, su aporte innovativo se encuentra en la propuesta de asistencia programable con características paramétricas y visuales.

Dado de que la programación, a través del planteamiento de algoritmos eficientes, ha sido de gran apoyo en la ingeniería eléctrica ya que ha destacado por solucionar problemas en distintos enfoques temáticos; en la presente tesis se busca la aplicación de herramientas programadas que asistan al profesional en tareas de diseño y cálculo de instalaciones eléctricas.

Como parte fundamental, la mencionada aplicación se basa en hallarse en una plataforma la cual se caracteriza por ser moderna como lo es la metodología del *Building Information Modelling* o BIM, misma que a nivel nacional se ha vuelto un inconveniente pese a sus numerosas ventajas sobre la metodología y software predecesor.

En ese sentido, esta tesis detalla las posibilidades programables para implementar BIM en el desarrollo del diseño y cálculo de instalaciones eléctricas interiores, trayendo consigo una propuesta que sea aplicable a retos arquitectónicos convencionales a los que ingeniero electricista se enfrenta como parte de su trabajo.

Atentamente

Br. Luis Giuseppe Yucra Altamirano



## ÍNDICE GENERAL

Índice de figuras . . . . .	11
Índice de tablas . . . . .	17
Índice de algoritmos . . . . .	18
Índice de códigos . . . . .	19
Acrónimos . . . . .	20
<b>1 ASPECTOS GENERALES . . . . .</b>	<b>21</b>
1.1 Descripción del problema . . . . .	21
1.2 Formulación del Problema . . . . .	25
1.2.1 Problema General . . . . .	25
1.2.2 Problemas Específicos . . . . .	25
1.3 Objetivos del Estudio . . . . .	25
1.3.1 Objetivo General . . . . .	25
1.3.2 Objetivos Específicos . . . . .	25
1.4 Hipótesis . . . . .	26
1.4.1 Hipótesis General . . . . .	26
1.4.2 Hipótesis Específicas . . . . .	26
1.5 Variables . . . . .	26
1.5.1 Variable Independiente . . . . .	27
1.5.2 Variable Dependiente . . . . .	27
1.6 Justificación . . . . .	30
1.6.1 Justificación Técnica . . . . .	30
1.6.2 Justificación Social . . . . .	30
1.6.3 Justificación Económica . . . . .	31
1.6.4 Justificación Académica . . . . .	32
1.7 Alcances y Limitaciones . . . . .	33
1.7.1 Alcances . . . . .	33
1.7.2 Limitaciones . . . . .	33
1.8 Metodología . . . . .	34

1.8.1	La Investigación . . . . .	35
2	FUNDAMENTO TEÓRICO . . . . .	38
2.1	Introducción . . . . .	38
2.2	Antecedentes . . . . .	38
2.2.1	Antecedentes Locales . . . . .	38
2.2.2	Antecedentes Nacionales . . . . .	39
2.2.3	Antecedentes Internacionales . . . . .	39
2.3	Marco Normativo . . . . .	41
2.4	Marco Conceptual . . . . .	41
2.5	Marco Teórico . . . . .	43
2.5.1	Diseño y Cálculo de Instalaciones Eléctricas Interiores . . . . .	43
2.5.2	Modelado de Información de Construcción o <i>BIM</i> . . . . .	44
2.5.3	Software BIM aplicado de la Ingeniería Eléctrica . . . . .	47
2.5.4	Programación Paramétrica-Visual . . . . .	49
3	ANÁLISIS DE CARACTERÍSTICAS DE SOFTWARE BIM PARA LA ETAPA DE DI- SEÑO & CÁLCULO DE INSTALACIONES ELÉCTRICAS . . . . .	61
3.1	Introducción . . . . .	61
3.2	Instalaciones Eléctricas . . . . .	61
3.2.1	Consideraciones de la Especialidad . . . . .	62
3.2.2	Criterios para el Diseño y Cálculo . . . . .	62
3.2.3	Flujo de Trabajo Convencional . . . . .	64
3.3	Características BIM . . . . .	66
3.3.1	Necesidad . . . . .	66
3.3.2	Información . . . . .	67
3.3.3	Gestión . . . . .	68
3.4	Flujo de Trabajo BIM como Algoritmo . . . . .	70
3.5	Necesidad, Información & Gestión en Software BIM . . . . .	73
3.5.1	Elección del Software BIM . . . . .	73
3.5.2	Revit . . . . .	75
4	EVALUACIÓN DE LOS POTENCIALES DE LA PROGRAMACIÓN PARAMÉTRICA- VISUAL EN BIM . . . . .	94
4.1	Introducción . . . . .	94
4.2	Relación con el Software BIM . . . . .	94
4.3	Dynamo y su uso convencional . . . . .	95

4.3.1	Estilo de Programación . . . . .	98
4.3.2	Requerimientos de Información . . . . .	99
4.4	Potenciales de la aplicación de Dynamo . . . . .	102
4.4.1	Diseño . . . . .	102
4.4.2	Cálculo . . . . .	103
4.5	Marco Previo para el Desarrollo de Módulos . . . . .	105
5	DESARROLLO DE MÓDULOS . . . . .	107
5.1	Introducción . . . . .	107
5.2	Recursos Adicionales . . . . .	107
5.3	Modelado . . . . .	108
5.3.1	El Boceto Eléctrico a Programar . . . . .	108
5.3.2	Criterio profesional inicial . . . . .	110
5.3.3	Recursos Adicionales . . . . .	110
5.3.4	Alimentación Eléctrica . . . . .	111
5.3.5	Iluminación . . . . .	127
5.3.6	Tomacorrientes . . . . .	149
5.3.7	Corriente Débil . . . . .	157
5.4	Cálculo . . . . .	157
5.4.1	Base del Algoritmo . . . . .	158
5.4.2	Desarrollo del Script “Cargas” . . . . .	167
5.4.3	Desarrollo del Script “Caídas de Tensión” . . . . .	184
6	APLICACIÓN DE LA PROPUESTA . . . . .	191
6.1	Introducción . . . . .	191
6.2	Contexto Aplicativo . . . . .	191
6.3	Base Arquitectónica . . . . .	191
6.3.1	Características . . . . .	192
6.4	Ejecución de Módulos . . . . .	192
6.4.1	Consideraciones . . . . .	192
6.4.2	Flujo de Trabajo Propuesto . . . . .	194
6.4.3	Planteado de Circuitos . . . . .	197
6.4.4	Visualización de Conductos . . . . .	210
6.4.5	Ejecución de Cálculos . . . . .	237
6.5	Resultados . . . . .	251
6.5.1	Consideraciones . . . . .	251

6.5.2	Tiempos de ejecución . . . . .	251
6.5.3	Concordancia BIM . . . . .	252
6.5.4	Concordancia normativa . . . . .	252
6.6	Discusión . . . . .	253
CONCLUSIONES . . . . .		254
RECOMENDACIONES . . . . .		255
BIBLIOGRAFIA . . . . .		257
ANEXOS . . . . .		262

## ÍNDICE DE FIGURAS

Figura 1.1	Aplicaciones del BIM . . . . .	22
Figura 1.2	Grado de Adopción BIM . . . . .	23
Figura 1.3	Curva de Esfuerzo-Tiempo BIM . . . . .	32
Figura 2.1	LOD como Nivel de Detalle . . . . .	46
Figura 2.2	Ejemplo del Nivel de Desarrollo de un Objeto . . . . .	47
Figura 2.3	Interfaz de Programación Paramétrica Visual . . . . .	50
Figura 2.4	Comparativa de los entornos tradicional, visual y visual-paramétrico de programación . . . . .	52
Figura 2.5	Interfaz tradicional vs. visual . . . . .	52
Figura 2.6	Interfaz de <i>Scratch</i> como entorno visual de programación con fines educativos. . . . .	54
Figura 2.7	Copiamiento visual en diferentes entornos de programación . . . . .	55
Figura 2.8	Grasshopper 3D . . . . .	56
Figura 2.9	Interoperatividad entre Grasshopper 3D y Revit 2022 . . . . .	57
Figura 2.10	Uso paralelo de Rhinoceros, Grasshopper y Revit . . . . .	58
Figura 2.11	Python Software . . . . .	59
Figura 2.12	Dynamo Software . . . . .	60
Figura 3.1	Comparativa 2D, 3D & BIM . . . . .	69
Figura 3.2	<i>Clash Detection</i> en BIM. . . . .	70
Figura 3.3	<i>Workflow</i> BIM de Chen et al. para la especialidad eléctrica (Adaptado) . . . . .	72
Figura 3.4	Características BIM en el <i>Workflow</i> BIM de Chen et al. (Adaptado) . . . . .	72
Figura 3.5	Necesidad, Información & Gestión en Software BIM . . . . .	73
Figura 3.6	Autodesk Revit Software . . . . .	76
Figura 3.7	Interfaz de Revit como producto integral . . . . .	76
Figura 3.8	El entorno <i>Electrical</i> de Revit . . . . .	77
Figura 3.9	Configuración Adicional del Entorno <i>Electrical</i> . . . . .	77
Figura 3.10	Manejo de Unidades para disciplina Electrica en Revit. . . . .	78
Figura 3.11	Uso “Doble” Convencional Ilustrado . . . . .	79
Figura 3.12	Representación tridimensional de Modelo Complejo en Revit . . . . .	80
Figura 3.13	Vistas Auxiliares para el Modelado de Instalaciones Eléctricas en Revit . . . . .	81
Figura 3.14	Vistas de Revit para especialidades . . . . .	84
Figura 3.15	Jerarquía Informativa dentro de Revit . . . . .	85

Figura 3.16	Ficha de Propiedades de un elemento de Revit . . . . .	86
Figura 3.17	Funciones de Gestión dentro de Revit . . . . .	88
Figura 3.18	Punto de Conexión Eléctrica en Revit . . . . .	89
Figura 3.19	Interfaz para el cálculo de cargas en Revit. . . . .	90
Figura 3.20	Autodesk Navisworks Manage Software . . . . .	91
Figura 3.21	Ventana de Clash Detection de Navisworks . . . . .	92
Figura 4.1	Operación de Dynamo con fines arquitectónicos y estructurales . . . . .	96
Figura 4.2	Otros usos para Dynamo . . . . .	97
Figura 4.3	Sistema de Nodos, Enlaces, Entradas y Respuestas de <i>Dynamo</i> . . . . .	98
Figura 4.4	Procesos recursivos en <i>Dynamo</i> . . . . .	99
Figura 4.5	Implementación de procesos recursivos en <i>Dynamo</i> . . . . .	100
Figura 4.6	Manejo de información de un elemento eléctrico en <i>Dynamo</i> . . . . .	101
Figura 4.7	Ubicación y emplazamiento de elementos en <i>Dynamo</i> . . . . .	102
Figura 4.8	Manipulación de conectores y extracción de información de un elemento en <i>Dynamo</i> . . . . .	103
Figura 4.9	Nodos por defecto de <i>Dynamo</i> para la manipulación de tablas dentro de Revit	104
Figura 4.10	Dirroots Pluggin BIM . . . . .	105
Figura 5.1	Boceto eléctrico a programar N°1 – Alimentación . . . . .	109
Figura 5.2	Boceto eléctrico a programar N°2 – Iluminación . . . . .	109
Figura 5.3	Dialux EVO 10.1 Software . . . . .	110
Figura 5.4	AutoCAD 2022 Software . . . . .	111
Figura 5.5	Diagrama de Flujo para el script “Conductos-i” . . . . .	112
Figura 5.6	Interfaz para la selección de datos. . . . .	118
Figura 5.7	Mapeado de conectores por elemento seleccionado . . . . .	119
Figura 5.8	Implementación del Algoritmo 5.1 en Dynamo. . . . .	120
Figura 5.9	Traslación en ejes <i>Z</i> como traducción de montantes eléctricas. . . . .	122
Figura 5.10	Generación de conductos eléctricos basados en líneas geométricas. . . . .	123
Figura 5.11	Vista completa del script <i>Conductos-i</i> . . . . .	125
Figura 5.12	Diagrama de Flujo para el script “Iluminación Óptima” . . . . .	128
Figura 5.13	Diagrama de Flujo para el instalaciones de cajas de conexiones y entubado respectivo . . . . .	129
Figura 5.14	Diagrama referencial para entubado de cajas de conexión para circuitos de iluminación . . . . .	130
Figura 5.15	Configuración de nuevo perfil normativo en Dialux EVO . . . . .	132

Figura 5.16	Generación de nuevos perfiles en base a la norma <i>EM.010</i> del Reglamento Nacional de Edificaciones (RNE) . . . . .	133
Figura 5.17	Aviso sobre los modelos <i>.ifc</i> dentro de Dialux EVO . . . . .	134
Figura 5.18	Configuración básica para la exportación de datos de luminarias a formato <i>.dwg</i>	135
Figura 5.19	Mapeado de Luminarias por capas en AutoCAD . . . . .	136
Figura 5.20	Generación de archivo de texto con coordenadas de luminarias . . . . .	136
Figura 5.21	Extracción de coordenadas <i>XYZ</i> de luminarias hacia listas Dynamo . . . . .	137
Figura 5.22	Toma de datos para las luminarias y su respectiva capa generada por Dialux EVO . . . . .	138
Figura 5.23	Generación de puntos geométricos para luminarias por capa de Dialux EVO .	139
Figura 5.24	Métodos de emplazamiento de luminarias en <i>Dynamo</i> . . . . .	141
Figura 5.25	Instalación de cajas de conexión y empalmes para luminarias emplazadas . . .	142
Figura 5.26	Entrada de datos y filtro de conectores para el script <i>Conductos I</i> . . . . .	143
Figura 5.27	Implementación de los algoritmos 5.5 y 5.6 en <i>Dynamo</i> . . . . .	145
Figura 5.28	Traslación <i>Z</i> para la generación de líneas sin uso de programación tradicional	146
Figura 5.29	Generación de tubos a partir de líneas para sistemas de alumbrado . . . . .	147
Figura 5.30	Vista completa del script <i>Conductos-I</i> . . . . .	148
Figura 5.31	Diagrama referencial para el emplazamiento de cajas de conexión y entubado de circuitos de tomacorrientes . . . . .	150
Figura 5.32	Interfaz de selección de datos para el emplazamiento de cajas de conexión para tomacorrientes . . . . .	153
Figura 5.33	Instalación de cajas de conexión por puntos geométricos de tomacorrientes . .	153
Figura 5.34	Detección de conectores con vector direccional $-1$ para sistemas de tomacorrientes . . . . .	154
Figura 5.35	Implementación de los algoritmos 5.7 y 5.8 en <i>Dynamo</i> . . . . .	155
Figura 5.36	Traslación de puntos <i>Z</i> para sistemas de tomacorrientes . . . . .	156
Figura 5.37	Diagrama de Flujo para el cálculo de cargas . . . . .	159
Figura 5.38	Máximas caídas de tensión permitidas en un circuito . . . . .	164
Figura 5.39	Diagrama de flujo para cálculo de caídas de tensión . . . . .	165
Figura 5.40	Interfaz de selección de Tipo de Edificación según alcances del Código Nacional de Electricidad (CNE)-U . . . . .	168
Figura 5.41	Filtro para áreas según selección del tipo de edificación . . . . .	170
Figura 5.42	Interfaz para la toma de áreas y cargas eléctricas adicionales . . . . .	171
Figura 5.43	Adición normativa del Código Nacional de Electricidad (CNE)-U a la toma de datos por tipo de edificación. . . . .	172

Figura 5.44	Ordenamiento de Cargas Eléctricas Adicionales y de tipo Especial . . . . .	174
Figura 5.45	Cálculo de la Carga Básica para Viviendas Unifamiliares y Departamentos en Dynamo . . . . .	175
Figura 5.46	Cálculo de Carga Adicional, Cocina Eléctrica y Aire Acondicionado o Cale- facción para Viviendas Unifamiliares y Departamentos en Dynamo . . . . .	176
Figura 5.47	Cálculo de la Carga para Escuelas y Hospitales en Dynamo . . . . .	178
Figura 5.48	Cálculo de la Carga para Hoteles en Dynamo . . . . .	179
Figura 5.49	Cálculo de la Carga según <i>Tabla 14</i> en Dynamo . . . . .	181
Figura 5.50	Cálculo de Cargas Adicionales de uso NO doméstico en Dynamo . . . . .	182
Figura 5.51	Generación del Cuadro de Cargas en Dynamo . . . . .	183
Figura 5.52	Interfaz para la selección de tableros y conductos para el cálculo de caídas de tensión . . . . .	185
Figura 5.53	Extracción de datos eléctricos de circuitos para el cálculo de caídas de tensión	186
Figura 5.54	Obtención de longitud total por tramos de circuitos eléctricos asociados a sus respectivos conductos . . . . .	187
Figura 5.55	Aplicación del Algoritmo 5.12 para el desarrollo del script “Caídas de Tensión”	188
Figura 5.56	Generación de rango de colores en función a las caídas de tensión obtenidas para visualización en modelo BIM . . . . .	189
Figura 5.57	Generación del cuadro de caídas de tensión por tablero y circuito . . . . .	190
Figura 6.1	Base Arquitectónica BIM 3D para modelado de instalaciones eléctricas . . . . .	193
Figura 6.2	<i>Workflow</i> BIM para la especialidad de instalaciones eléctricas asistida por programación paramétrica visual, basado en el <i>Workflow</i> de Chen et al. (2016)	195
Figura 6.3	Diagrama de flujo para la ejecución de scripts . . . . .	196
Figura 6.4	Emplazamiento Manual de Medidores, Cajas de Paso y Tableros Eléctricos . . . . .	198
Figura 6.5	Migración de modelo BIM a Dialux por medio de un formato <i>.ifc</i> . . . . .	199
Figura 6.6	Diseño lumínico BIM asistido por Dialux con base de formato <i>.ifc</i> . . . . .	201
Figura 6.7	Posiciones provistas por Dialux para el procesamiento en AutoCAD . . . . .	202
Figura 6.8	Coordenadas de luminarias extraídas de Dialux por medio del script de Thar- wat (2010) . . . . .	203
Figura 6.9	Igualdad de luminarias BIM en Dialux y Revit . . . . .	203
Figura 6.10	Interfaz de ejecución del script “Iluminación Óptima” . . . . .	204
Figura 6.11	Comparativa BIM tras la ejecución del script “Iluminación Óptima” . . . . .	205
Figura 6.12	Tiempo de ejecución del script “Iluminación Óptima” . . . . .	206
Figura 6.13	Tiempo de ejecución por nodo del script “Iluminación Óptima” . . . . .	207
Figura 6.14	Generación del circuito eléctrico por medio de herramientas BIM. . . . .	209



Figura 6.15	Datos BIM de tipo eléctrico–cargas generados para un tablero eléctrico. . . . .	210
Figura 6.16	Modelo de Dialux con diseño lumínico para ser reutilizado en Software BIM . . . . .	212
Figura 6.17	Planteado Automático de Circuitos de Tomacorrientes por medio de herramientas del entorno <i>Electrical</i> del Software BIM . . . . .	213
Figura 6.18	Planteado Automático de Circuitos de Alumbrado por medio de herramientas del entorno <i>Electrical</i> del Software BIM . . . . .	214
Figura 6.19	Interfaz de ejecución de script para generación de conductos de alimentadores. . . . .	215
Figura 6.20	Generación automática de entubado y montante eléctrica . . . . .	216
Figura 6.21	Tiempo de ejecución del script “Conductos-I” . . . . .	217
Figura 6.22	Tiempo de ejecución por nodo del script “Conductos-I” . . . . .	218
Figura 6.23	Interfaz agregada para la inclusión de cajas de conexión en circuitos de alumbrado. . . . .	219
Figura 6.24	Interfaz agregada para la selección de tipo de caja de conexión en circuitos de alumbrado. . . . .	220
Figura 6.25	Emplazamiento de cajas de conexión precisadas por coordenadas precisas de luminarias. . . . .	221
Figura 6.26	Tiempo de ejecución del script “Iluminación Óptima” más emplazamiento de cajas de conexión . . . . .	222
Figura 6.27	Tiempo de ejecución por nodo del script “Iluminación Óptima” más emplazamiento de cajas de conexión . . . . .	223
Figura 6.28	Interfaz para el emplazamiento de cajas rectangulares. . . . .	224
Figura 6.29	Tiempo de ejecución del script “Tomacorrientes” más emplazamiento de cajas de conexión . . . . .	225
Figura 6.30	Tiempo de ejecución por nodo del script “Tomacorrientes” más emplazamiento de cajas de conexión . . . . .	225
Figura 6.31	Vistas múltiples de un mismo proyecto como apoyo a la ejecución de scripts dentro del software BIM. . . . .	226
Figura 6.32	Propuestas de zona troncal de circuitos de iluminación y tomacorrientes . . . . .	228
Figura 6.33	Interfaz de ejecución del script para entubados por techo y suelo . . . . .	230
Figura 6.34	Vista del entubado generado por el script de entubado por techo. . . . .	231
Figura 6.35	Vista del entubado generado por el script de entubado por suelo. . . . .	232
Figura 6.36	Tiempo de ejecución por nodo de scripts de conductos por techo y suelo. . . . .	234
Figura 6.37	Tiempo de ejecución por nodo de scripts de conductos por techo y suelo. . . . .	235
Figura 6.38	Aplicación de <i>Dynamo</i> para la toma de longitudes de conductos generados. . . . .	236
Figura 6.39	Extracción de datos de áreas por medio de herramientas BIM. . . . .	237

Figura 6.40	Interfaz de generación de cuadro de cargas basados en tipos de edificación de la Sección 050 del Código Nacional de Electricidad (CNE)–U . . . . .	238
Figura 6.41	Interfaz de generación de cuadro de cargas basados en tipos de edificación de la Tabla N° 14 del Código Nacional de Electricidad (CNE)–U . . . . .	238
Figura 6.42	Interfaz para la toma de datos de área y posibles cargas adicionales . . . . .	239
Figura 6.43	Cuadro de Cargas generado para un local comercial . . . . .	240
Figura 6.44	Tiempo de ejecución del script “Cuadro de Cargas” . . . . .	240
Figura 6.45	Tiempo de ejecución por nodo del script “Cuadro de Cargas” . . . . .	241
Figura 6.46	Cuadro de Cargas generado para una unidad de vivienda en edificio departamental . . . . .	242
Figura 6.47	Tiempo de ejecución del script “Cuadro de Cargas” . . . . .	242
Figura 6.48	Tiempo de ejecución por nodo del script “Cuadro de Cargas” . . . . .	243
Figura 6.49	Modelo BIM del entubado para un tablero de distribución previo a la ejecución del script de cálculo de caídas de tensión . . . . .	244
Figura 6.50	Diferencia de parámetros del calibre de conductor . . . . .	245
Figura 6.51	Interfaz para el cálculo de caídas de tensión . . . . .	246
Figura 6.52	Cuadro de Caídas de Tensión generado por script . . . . .	246
Figura 6.53	Vista tridimensional generada con rango de colores para interpretación de Caídas de Tensión . . . . .	247
Figura 6.54	Tiempo de ejecución del script “Caídas de Tensión” . . . . .	248
Figura 6.55	Tiempo de ejecución por nodo del script “Caídas de Tensión” . . . . .	249
Figura 6.56	Recálculo de caídas de tensión con el calibre automático de Revit . . . . .	250

## ÍNDICE DE TABLAS

Tabla 1.1	Reporte de Obras Paralizadas . . . . .	24
Tabla 1.2	Matriz de operacionalización de variables . . . . .	29
Tabla 1.3	Matriz de Consistencia . . . . .	37
Tabla 2.1	Dimensiones del BIM . . . . .	46
Tabla 2.2	LOD en el BIM . . . . .	47
Tabla 2.3	Comparativa Programación tradicional vs. Programación paramétrica-visual.	53
Tabla 3.1	Especialidad Eléctrica en Software BIM . . . . .	74
Tabla 3.2	Comparativa Software BIM . . . . .	75
Tabla 3.3	Errónea Traducción Literal. Caso: Corriente Eléctrica . . . . .	79
Tabla 3.4	Capacidad de tener Características Premodeladas en Revit . . . . .	82
Tabla 5.1	Campos de un Cuadro de Cargas según Ministerio de Energía y Minas (2008)	159
Tabla 5.2	Resumen de consideraciones para el cálculo de cargas según Código Nacional de Electricidad (CNE)-U . . . . .	160
Tabla 5.3	Tipos de Área para el cálculo por potencia superficial ( $W/m^2$ ) . . . . .	177
Tabla 6.1	Resultados de tiempos de ejecución . . . . .	251
Tabla 6.2	Resultados BIM . . . . .	252
Tabla 6.3	Resultados de concordancia normativa . . . . .	252

## ÍNDICE DE ALGORITMOS

5.1	Cálculo del conector más próximo y generación de líneas . . . . .	113
5.2	Función <i>Lin2</i> , parte del Algoritmo 5.1 . . . . .	115
5.3	Traslación en eje <i>Z</i> para los puntos inicial y final de <i>LIN</i> . . . . .	116
5.4	Funciones <i>TransInf</i> y <i>TransSup</i> , parte del Algoritmo 5.3 . . . . .	116
5.5	Cálculo del conector más próximo y generación de líneas para sistemas de iluminación	130
5.6	Función <i>Lin2_I</i> , parte del Algoritmo 5.5 . . . . .	131
5.7	Cálculo del conector más próximo y generación de líneas para sistemas de tomacorrientes	151
5.8	Función <i>Lin2_T</i> , parte del Algoritmo 5.7 . . . . .	152
5.9	Cálculo de Carga Básica para Viviendas Unifamiliares y Edificios Departamentales . .	161
5.10	Creación de Matriz de Datos para Cargas Adicionales . . . . .	162
5.11	Cálculo de Cargas Adicionales por presencia de cocinas eléctricas . . . . .	163
5.12	Cálculo de Caídas de Tensión y Generación de Matriz de resultados . . . . .	166

## ÍNDICE DE CÓDIGOS

Código A.1	Implementación de los Algoritmos 1 y 2 . . . . .	262
Código A.2	Implementación de los Algoritmos 3 y 4 . . . . .	265
Código A.3	Implementación de los Algoritmos 5 y 6 . . . . .	266
Código A.4	Implementación de los Algoritmos 7 y 8 . . . . .	268
Código A.5	Implementación del Algoritmo 9 . . . . .	270
Código A.6	Implementación del Algoritmo 11 . . . . .	271
Código A.7	Implementación del Algoritmo 10 . . . . .	272
Código A.8	Implementación del Algoritmo 12 . . . . .	273
Código A.9	Herramienta de Tharwat (2010) . . . . .	274

## ACRÓNIMOS

<b>BIM</b>	Building Information Modeling
<b>CAD</b>	Computer-Aided Design
<b>CNE</b>	Código Nacional De Electricidad
<b>DGE</b>	Dirección General De Electricidad
<b>HVAC</b>	Heating, Ventilation And Air Conditioning
<b>LOD</b>	Level Of Development
<b>MEF</b>	Ministerio De Economía Y Finanzas
<b>MINEM</b>	Ministerio De Energía Y Minas
<b>MVCS</b>	Ministerio De Vivienda, Construcción Y Saneamiento
<b>NTP</b>	Normas Técnicas Peruanas
<b>RNE</b>	Reglamento Nacional De Edificaciones
<b>SLUMP</b>	Sistema Legal De Unidades De Medida Del Perú
<b>Wf</b>	Workflow

## CAPÍTULO I

### ASPECTOS GENERALES

#### 1.1 Descripción del problema

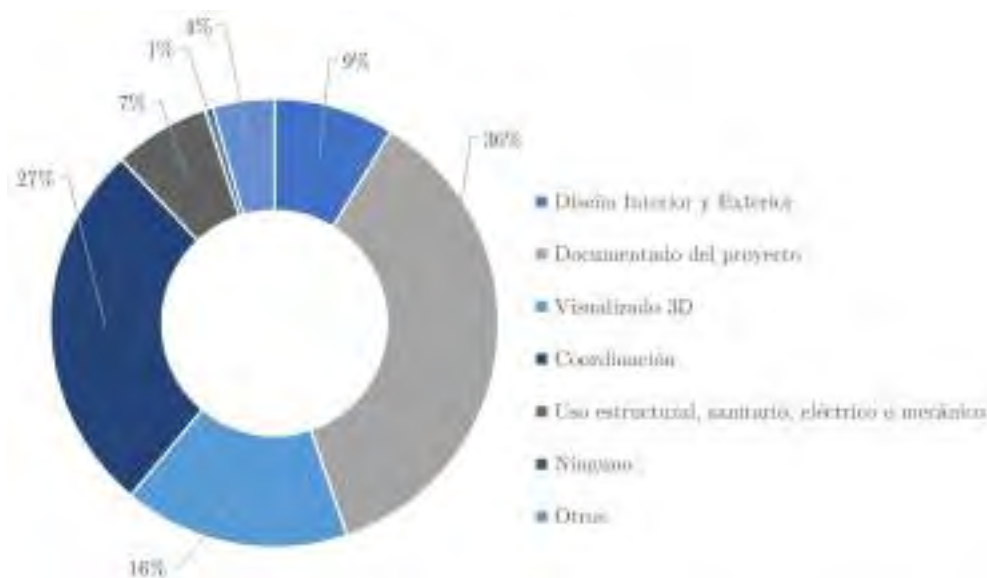
Llegado el año 2022, el concepto de Building Information Modeling (BIM) y los softwares asociados a esta metodología ya no representan una novedad en el ámbito de la ingeniería relacionada con el sector de la construcción e instalaciones eléctricas. Diversos países europeos, asiáticos y americanos, como España, Alemania, China, Japón, Singapur, Brasil e incluso Chile, han implementado un Plan BIM desde hace varios años (Almeida, 2019). En cambio, en el Perú, fue recién en el año 2019 cuando se estableció un marco legal para su aplicación en proyectos del sector público, junto con una hoja de ruta que busca impulsar su adopción en los próximos años. Es relevante mencionar que, de manera independiente y a menor escala, el sector privado ha ido adoptando gradualmente esta metodología, aunque aún enfrenta cierta resistencia. Según una investigación realizada en el año 2021, en la ciudad de Lima y la provincia constitucional del Callao, el 60.9% de los proyectos privados no habían implementado el BIM (Murguía Sánchez et al. 2021, p. 10).

A pesar de que en sus inicios, el BIM fue considerado como la solución y una mejora respecto a las limitaciones del Computer-Aided Design (CAD); hoy en día, dentro del sector empresarial especializado, se le considera como una solución parcial, ya que aún presenta dificultades en su uso e implementación.

Dentro de los principales factores inconvenientes en la implementación del BIM, se halla la elección correcta del software para su uso.

En la actualidad, hay una amplia variedad de desarrolladores de software BIM en el mundo, cada uno con su propia dinámica que cumple con los lineamientos metodológicos del BIM. No obstante, debido a la complejidad y al extenso catálogo de configuraciones disponibles, especialmente al trabajar con la especialidad de instalaciones eléctricas, se tiende a invertir tiempo adicional en la capacitación y entrenamiento para el uso del software seleccionado, lo cual retrasa el desarrollo del proyecto final.

La complejidad del software BIM puede ser vista de manera subjetiva, y se espera que se proponga una mejora continua mediante desafíos y escenarios innovadores. Sin embargo, al compararse con la metodología anteriormente utilizada (CAD) y su simplicidad, los ingenieros electricistas y otros profesionales de la especialidad encuentran resistencia en migrar hacia esta nueva metodolo-

**Figura 1.1***Aplicaciones del BIM*

*Fuente:* Adaptado a partir de Vizcarra Aparicio (2018, p. 9)

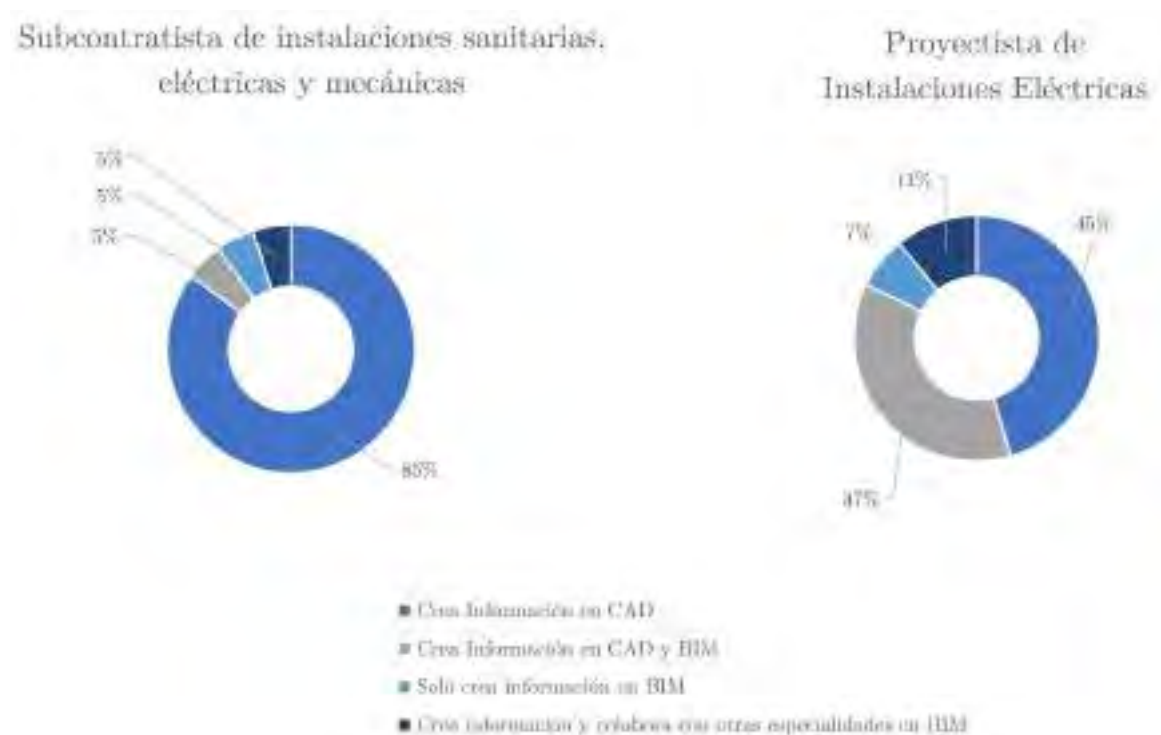
gía, a pesar de las numerosas ventajas que el BIM ofrece. De hecho, solo un 7% de los ingenieros especializados en instalaciones y estructuras han adoptado y aplicado el BIM, según los resultados de un estudio realizado por Vizcarra Aparicio (2018) (ver Figura 1.1).

También se debe destacar que, vista en la Figura 1.2, data extraída de la investigación de Murguía Sánchez et al. (2021), la cual ofrece una perspectiva más dirigida hacia la especialidad de instalaciones eléctricas, esta evidencia que aún se tiene dependencia total o parcial sobre el CAD, ya que, dentro de los subcontratistas, más de la mitad de la muestra correspondiente, aún permanece con el CAD; mientras que proyectistas se ubican con un alto 45%.

A corto plazo, si persiste el escepticismo hacia el BIM debido a su complejidad, manteniendo así el estado actual y descartando su implementación, continuarán las deficiencias en la elaboración de expedientes técnicos, lo cual ha sido identificado como la principal causa de obras paralizadas en el año 2022, según la Contraloría General de la República del Perú (Ver Tabla 1.1). En cambio, a largo plazo, como nación, se verá afectado por un atraso técnico-profesional en comparación con otros países.

A pesar de los beneficios que ofrece la metodología BIM, su complejidad se considera un inconveniente, especialmente bajo esquemas arquitectónicos convencionales. Si se suma a esta problemática la excentricidad en el diseño de edificaciones de los últimos tiempos, el modelado del proyecto en un software que no se limite únicamente al dibujo vectorial (CAD) y que requiera información adicional para su interpretación por parte de profesionales, se vuelve aún más complicado.



**Figura 1.2***Grado de Adopción BIM*

*Fuente:* Adaptado a partir de Murguía Sánchez et al. (2021, p. 17)

En el contexto de la especialidad de instalaciones eléctricas, tanto en arquitecturas minimalistas como excéntricas, el modelado de conductos y circuitos eléctricos, incluyendo instalaciones de corrientes débiles, demanda un tiempo considerablemente alto. La cantidad de información necesaria para cumplir con la metodología BIM aumenta significativamente cuando se deben adaptar a las necesidades arquitectónicas y sociales de un proyecto complejo. Esto contribuye a la problemática mencionada anteriormente en cuanto a la dificultad de uso del software enfocado en BIM.

En el ámbito nacional, la globalización y el hecho de que la mayoría de las empresas desarrolladoras de software BIM sean extranjeras, ha llevado a la necesidad de adaptarse al uso de programas que no se ajustan completamente a la normativa peruana vigente. Estas empresas, en un esfuerzo por vender productos compatibles con distintos países, ofrecen software con preconfiguraciones basadas en marcos normativos estadounidenses y europeos. Esto amplía la gama y complejidad de opciones y aspectos que requieren una configuración adicional por parte del ingeniero electricista, con el objeto de cumplir estrictamente con las normas nacionales de diseño y cálculo de instalaciones eléctricas.

El entorno problemático expuesto es de gran consideración, ya que provoca demoras indeseables durante la etapa de diseño y cálculo de instalaciones eléctricas en proyectos de edificación que adoptan la metodología BIM. Además, estas demoras afectan a una escala mayor, ya que repercuten

**Tabla 1.1***Reporte de Obras Paralizadas*

<b>Causas de la paralización</b>	<b>Cantidad de Obras</b>
Asociado a deficiencias de expediente técnico	49
Asociado a deficiente supervisión de obra	47
Asociado a la calidad de obra	41
Asociado al Plan COVID 19	32
Aprobaciones sin sustento	29
Ausencia de personal clave en obra	27
Partidas no ejecutadas	20
Inadecuado almacén de materiales	17
No registran Infobras	16
Inexistencia de cuaderno de obra	14
Problemas con el terreno	11
Falta de equipamiento técnico	10
Asociado a cartas fianzas y garantías	8
Asociado a la calidad de los materiales	7
Asociado a la disponibilidad presupuestal	2
<i>Otros:</i>	
Subsanación de observaciones	
Atraso en trabajos realizados	
Asociado a la convocatoria	81
Asociado al reinicio de obra	
Asociado a coeficientes de la formula polinómica	

*Fuente:* Adaptado a partir de Contraloría General de la República del Perú (2022, p.29)

en el cronograma general del proyecto. Como resultado, el propietario o beneficiario del proyecto es el principal perjudicado ante cualquier tipo de prórroga o retraso en la entrega del mismo.

Aunque existe la posibilidad de mitigar estas demoras al obviar aspectos normativos en el software, esto conlleva el riesgo de que los proyectos sean parcialmente acordes a la normativa o, en el peor de los casos, estén en incumplimiento total.

En cuanto al incumplimiento normativo, este puede ser corregido mediante exhaustivos diagnósticos llevados a cabo por un comité técnico revisor. Sin embargo, esta misma acción puede derivar en una ramificación de problemas temporales, traducéndose en demora tras demora en la subsanación de errores. Coincidentemente, esta situación también es una de las causas de la paralización de obras mencionada previamente en la Tabla 1.1.

## 1.2 Formulación del Problema

### 1.2.1 *Problema General*

¿Será posible la creación de una herramienta informática basada en Programación Paramétrica-Visual aplicada a software BIM para el desarrollo de la etapa de diseño y cálculo de instalaciones eléctricas interiores?

### 1.2.2 *Problemas Específicos*

1. ¿Qué características tiene el software BIM para el diseño y cálculo de instalaciones eléctricas?
2. ¿Qué posibilidades ofrece la Programación Paramétrica-Visual dentro de la especialidad de instalaciones eléctricas interiores hechas en software BIM?
3. ¿Se podrán desarrollar los algoritmos, bloques y códigos que puedan resolver el funcionamiento de la herramienta para su aplicación en software BIM?
4. ¿Se podrá observar el funcionamiento de la propuesta por medio de su aplicación en una base arquitectónica?

## 1.3 Objetivos del Estudio

### 1.3.1 *Objetivo General*

Elaborar una herramienta informática basada en Programación Paramétrica-Visual aplicada a software BIM para el desarrollo de la etapa de diseño y cálculo de instalaciones eléctricas interiores.

### 1.3.2 *Objetivos Específicos*

1. Determinar las características que posee el software BIM para el diseño y cálculo de instalaciones eléctricas.
2. Hallar las posibilidades de la Programación Paramétrica-Visual para su aplicación en la especialidad de instalaciones eléctricas de tipo interiores.
3. Desarrollar algoritmos, bloques y códigos que expliquen el funcionamiento de los módulos para su aplicación en software BIM.

4. Aplicar la propuesta a una base arquitectónica en sus alcances de diseño y cálculo de instalaciones eléctricas interiores.

## 1.4 Hipótesis

### 1.4.1 *Hipótesis General*

La propuesta elaborada, teniendo consigo las posibilidades que ofrece la Programación Paramétrica-Visual cuando es aplicada al Software BIM, permite el desarrollo de la etapa de diseño y cálculo de instalaciones eléctricas interiores.

### 1.4.2 *Hipótesis Específicas*

1. Las características del software BIM para la etapa de diseño y cálculo de instalaciones eléctricas responden a la comprensión de necesidades, inyección de información necesaria y posibilidades de gestión del proyecto.
2. Se hallan posibilidades dentro de la Programación Paramétrica-Visual, sin embargo, se encuentran limitaciones de compatibilidad informática con el software BIM.
3. El desarrollo de módulos está en firme concordancia a un proceso algorítmico en el flujo de trabajo BIM para instalaciones eléctricas interiores.
4. La aplicación de la propuesta demuestra una facilidad de generación del modelo de instalaciones eléctricas interiores, estando acorde a las definiciones de la metodología BIM y a la norma vigente.

## 1.5 Variables

Para el presente plan de Tesis se considera la siguiente configuración de variables en cuanto a su dependencia.

### 1.5.1 *Variable Independiente*

#### 1.5.1.1 La Programación Paramétrica–Visual

La variable define a la base programable sobre la cual se pretende elaborar una propuesta. Las dimensiones que se hallan dentro de esta variable son:

- Compatibilidad informática
  - Software BIM compatible
- Método de Operación
  - Tipo de ejecución en software BIM

#### 1.5.1.2 El software BIM

La variable independiente a estudiar entiende la relación e inclusión de la metodología BIM al software sobre el cual se pretende ejecutar los módulos. Las dimensiones e indicadores que se hallan dentro de esta variable son:

- Complejidad del software
  - Especialidad eléctrica.
  - Cantidad de ajustes.
  - Comprende necesidades.
- Características arquitectónicas pre-modeladas
  - Tipo de vínculo informático.
  - Información modelada disponible.
- Capacidad de gestión de proyectos.
  - Coordinación interdisciplinaria disponible.

### 1.5.2 *Variable Dependiente*

#### 1.5.2.1 La etapa de diseño y cálculo de instalaciones eléctricas en BIM

Esta variable se describe como los aspectos gráficos y de cálculos propios de ingeniería aplicada a las instalaciones eléctricas interiores hechas en software BIM, y que se pueden obtener en base a la aplicación de los módulos. Sus dimensiones son:

- Facilidad de generación de modelo BIM.
  - Tiempo de ejecución de la propuesta.
- Concordancia BIM
  - Dimensiones BIM abarcadas.
- Concordancia con la normativa peruana.
  - Normas cumplidas.

**Tabla 1.2**

*Matriz de operacionalización de variables*

VARIABLE	TIPO	DEFINICIÓN	OPERACIONALIZACIÓN	DIMENSIONES	INDICADORES	INSTRUMENTO
Programación paramétrica visual	Independiente	Se entiende como una forma de conectar software alguno de modelado, el mismo que tenga capacidad informativa de manipulación de parámetros por medio de una interfaz visual de programación. (Kensek, 2014, p. 3)	La operación de la variable contempla la compatibilidad que pueda ofrecer para su ejecución e implementación, del mismo modo, su método de operación influye en la forma en cómo se pueda construir algoritmos.	Compatibilidad	Software BIM compatible	Manual de usuario
				Método de Operación	Tipo de ejecución	
Software BIM	Independiente	Programa informático sobre el cual se aplica la metodología BIM en sus alcances dimensionales de diseño y cálculo para distintos flujos de trabajo de un proyecto de instalaciones eléctricas interiores. (Demchak et al. 2009, p. 10)	La variable define su operación y su uso en principalmente la complejidad en su uso, las características que se pueden premodelar trayendo consigo la información pertinente, teniendo en cuenta su capacidad de gestión a través de la coordinación entre disciplinas.	Complejidad de Software	Especialidad eléctrica Cantidad de ajustes	Manual de usuario
				Características Arquitectónicas Pre-Modeladas	Tipo de vinculo informatico Información modelada disponible	Interfaz de software
				Capacidad de gestión de proyecto	Coordinación interdisciplinas disponible	Manual de usuario
La etapa de diseño y cálculo de instalaciones eléctricas en BIM	Dependiente	Es definido por el resultado obtenido por la aplicación de los algoritmos de la propuesta objetivo de la presente tesis.	La operación dentro de su naturaleza dependiente ofrece a la facilidad de generación de modelos BIM a partir de la elección del Software BIM y la interfaz de programación adecuada para el software. A su vez, estos modelos proyectuales se adhieren en concordancia al BIM y a la norma vigente.	Facilidad de generación de modelo BIM	Tiempo de ejecución	Interfaz de programación
				Concordancia BIM	Dimensiones BIM abarcadas	Referencias BIM
				Concordancia normativa	Normas cumplidas	Referencias normativas

*Fuente:* Elaboración propia.

## 1.6 Justificación

### 1.6.1 *Justificación Técnica*

Aún en el año 2022, el desafío del modelado de instalaciones en Software BIM sigue siendo considerable, a pesar de las actualizaciones anuales de los programas. Incluso en la investigación realizada por Mcfarland (2007, p. 36), en sus conclusiones para el año de su publicación, se destaca que los modeladores de instalaciones (ya sean mecánicas, sanitarias o eléctricas) suelen mostrar recelo hacia la metodología BIM debido a las dificultades en el modelado y la posterior documentación de sus proyectos.

En la última década, la compatibilidad entre las interfaces dedicadas a la programación paramétrica-visual y los software BIM y de modelado 3D ha ido en aumento, ganando popularidad entre los profesionales involucrados en el desarrollo BIM de proyectos. Sin embargo, esta herramienta ha sido ampliamente difundida en campos como el Diseño Generativo (*Generative Design*) para arquitectos, la mejora de flujos de trabajo para ingenieros estructurales, el cálculo de sistemas de calefacción, ventilación y aire acondicionado o Heating, Ventilation and Air Conditioning (HVAC) e incluso el modelado de redes sanitarias.

Es por esta razón que se justifica la elaboración de la presente tesis, con el objetivo de incluir a los ingenieros electricistas en el eficiente trabajo BIM y evitar caer en el arcaico pensamiento de “adaptar el software a sus necesidades”. En cambio, se busca migrar hacia un enfoque en el cual el ingeniero se adapte a las herramientas del software, aprovechando al máximo sus capacidades existentes.

### 1.6.2 *Justificación Social*

En el año 2022, el Plan BIM elaborado por el Ministerio de Economía y Finanzas (MEF) establece que los proyectos de inversión pública en edificaciones deben ser desarrollados bajo la metodología BIM. Por lo tanto, es necesario que los proyectos que se encuentran en la etapa de diseño (Modelado BIM 3D) sean más eficientes que aquellos que siguen métodos tradicionales. En este sentido, la correcta aplicación de una herramienta dedicada a una especialidad dentro de un trabajo que suele ser manual y arduo, generará resultados que también se hayan desarrollado bajo el enfoque BIM, pero con mayor rapidez y versatilidad.

Tanto para proyectos de carácter público como privado, los beneficiarios y propietarios podrán disfrutar de las edificaciones en mucho menos tiempo y con menos riesgo de posibles prórrogas o, en el peor de los casos, de paralización del proyecto. La implementación de una herramienta que



agilice y optimice la etapa de diseño y cálculo de instalaciones eléctricas interiores bajo la metodología BIM contribuirá a una mayor eficiencia en la ejecución de los proyectos, lo que beneficiará a la sociedad en general.

Además, la adopción de la metodología BIM y el uso de herramientas específicas para cada especialidad, como en el caso de instalaciones eléctricas, fomentará el avance tecnológico y la modernización de la industria de la construcción en el país. La incorporación de tecnologías y metodologías innovadoras mejora la calidad de los proyectos, la seguridad de las edificaciones y el impacto ambiental, lo que repercute positivamente en la comunidad.

Es por eso que, la implementación de la herramienta propuesta no solo permitirá cumplir con los requerimientos del Plan BIM y agilizar los procesos de diseño y cálculo de instalaciones eléctricas, sino que también contribuirá al desarrollo social al facilitar la construcción de edificaciones de manera más eficiente, segura y sostenible.

### 1.6.3 *Justificación Económica*

La Figura 1.3, que representa la “Curva de Esfuerzo–Tiempo BIM”, elaborada por MacLeamy (2008), muestra una comparativa gráfica entre una metodología tradicional y la metodología BIM a lo largo de las etapas clave de un proyecto, desde la concepción del diseño (*Predesign*) hasta la ejecución de la obra (*Construction*).

Además de esta comparativa, el gráfico incluye otras dos curvas relacionadas con el aspecto económico, aplicables tanto a la metodología tradicional como al BIM.

En el contexto de esta tesis, centrado en la metodología BIM, es importante realizar una evaluación económica específica para esta metodología. Se pueden destacar los siguientes aspectos:

Primero, el BIM tiene una mayor capacidad para impactar en los costos y la funcionalidad cuando se implementa en las etapas iniciales del proyecto.

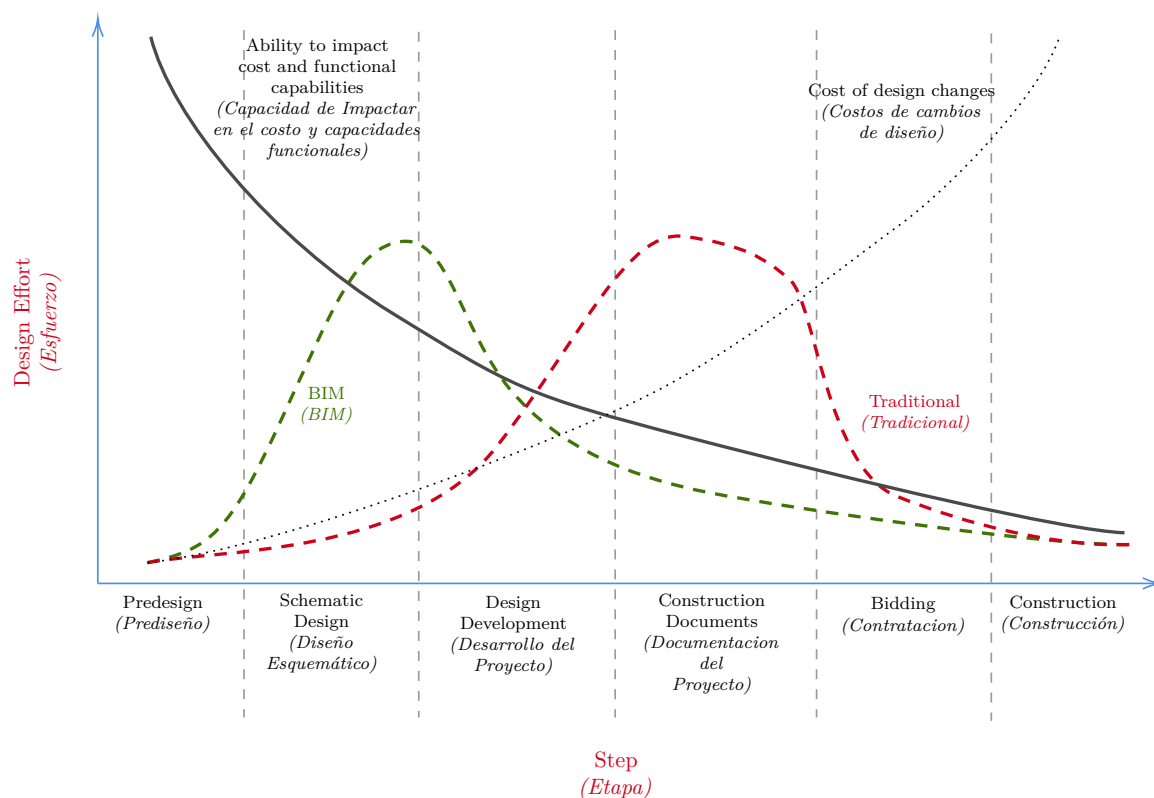
Y en segundo lugar, los costos asociados a cambios en el diseño son menores cuando se realizan en las primeras etapas del proyecto.

Si bien el BIM presenta una complejidad elevada, especialmente en las primeras etapas del diseño, es precisamente en este momento donde se pueden disfrutar de los beneficios económicos mencionados. Sin embargo, la problemática reside en la alta complejidad y el esfuerzo requerido en esta fase del proyecto.

Por lo tanto, si se pueden identificar oportunidades para reducir esa complejidad y el nivel de esfuerzo asociado, se lograría la mejor combinación en términos económicos para el proceso de proyección de una edificación. Esto significa que si se desarrolla una herramienta informática basada en Programación Paramétrica–Visual aplicada al Software BIM para la etapa de diseño y cálculo de

Figura 1.3

## Curva de Esfuerzo-Tiempo BIM



Fuente: Adaptado a partir de MacLeamy (2008)

instalaciones eléctricas interiores, se podría mejorar la eficiencia y reducir los costos asociados con el diseño y la planificación de las instalaciones eléctricas bajo la metodología BIM.

#### 1.6.4 Justificación Académica

A la fecha de redacción de este trabajo de tesis, tras una consulta bajo las palabras *BIM*, *Programación Paramétrica-Visual* e *Instalaciones Eléctricas*; en los repositorios académicos oficiales de la Universidad Nacional de San Antonio Abad del Cusco, como la Biblioteca Especializada<sup>1</sup> y el Repositorio Institucional UNSAAC<sup>2</sup>, se pudo constatar que no existen registros de proyectos o investigaciones que aborden la metodología BIM enfocada en la automatización de la metodología, específicamente en el ámbito de las instalaciones eléctricas.

Este escenario podría ser objeto de estudio en sí mismo para analizar las posibles razones detrás de esta falta de exploración en el campo del BIM en la universidad. Sin embargo, dado que el BIM es un campo ampliamente explorado en el ámbito internacional y que representa una importante

<sup>1</sup> <http://especializada.unsaac.edu.pe/>

<sup>2</sup> <http://repositorio.unsaac.edu.pe/>

tendencia en la ingeniería del siglo XXI, es fundamental que se desarrolle investigación y proyectos que busquen mejorar y optimizar las tecnologías actualmente utilizadas en el ámbito de la ingeniería eléctrica.

La implementación de una herramienta informática basada en Programación Paramétrica–Visual aplicada al Software BIM para la etapa de diseño y cálculo de instalaciones eléctricas interiores supone una oportunidad de crecimiento en las capacidades de ingeniería del país. Además, la introducción de esta herramienta puede despertar el interés de los estudiantes de la escuela profesional de Ingeniería Eléctrica de la Universidad Nacional de San Antonio Abad del Cusco, incentivándolos a mejorar sus competencias y habilidades en el campo de la metodología BIM.

La justificación académica de esta tesis radica en la necesidad de abordar un tema innovador y relevante en el contexto de la ingeniería eléctrica, como lo es la aplicación de la metodología BIM y la Programación Paramétrica–Visual en el diseño y cálculo de instalaciones eléctricas interiores. Esta investigación puede abrir nuevas oportunidades de estudio y desarrollo en la universidad, contribuyendo al crecimiento y avance de la ingeniería en el país.

## **1.7 Alcances y Limitaciones**

### **1.7.1 Alcances**

- Esta primera versión de la propuesta abarcará exclusivamente la especialidad de instalaciones eléctricas de redes en baja tensión de tipo interiores.
- Los resultados de la aplicación de la propuesta estarán enfocados en las características arquitectónicas específicas que se expongan en el respectivo capítulo.
- Los resultados obtenidos estarán en concordancia con el marco normativo correspondiente a instalaciones eléctricas interiores.
- La propuesta será compatible solamente con el software BIM que posea tal característica a la fecha de realización de la presente tesis.

### **1.7.2 Limitaciones**

- Los módulos de la propuesta funcionarán utilizando únicamente los recursos previamente instalados en el software BIM, sin agregar información adicional.

- La propuesta no desarrollará extensiones adicionales para el software BIM, limitándose a utilizar las funcionalidades ya presentes en su versión al momento de la investigación.
- El funcionamiento de la propuesta se limitará únicamente a la etapa de diseño y cálculo de las instalaciones eléctricas interiores. No se abordará la generación de costos y presupuestos, ya que existen softwares especializados para esta tarea.
- La propuesta no incorporará inteligencia artificial que implique el “auto-diseño” o la independencia del criterio profesional del ingeniero electricista usuario. Su uso estará siempre bajo la supervisión y decisión del ingeniero.
- Considerando que las interfaces de programación y el software BIM pueden estar en versiones de prueba o “beta” para el año 2023, no se tomarán en cuenta estas versiones preliminares, centrándose únicamente en productos finales y estables.

## 1.8 Metodología

La metodología aplicada se puede entender en el consecuente de las fases descritas a continuación.

1. **Revisión Normativa y Conceptual** En esta fase inicial, se lleva a cabo una revisión exhaustiva de los aspectos normativos que rigen el diseño y cálculo de las instalaciones eléctricas. Además, se realiza un estudio en ingeniería para comprender las bases conceptuales de la programación paramétrica-visual, la metodología BIM y el software BIM disponible, así como sus capacidades para integrar un proyecto de instalaciones eléctricas. El objetivo de esta fase es la obtención de información necesaria para desarrollar las etapas posteriores.
2. **Análisis de las características del software BIM** En esta etapa, se analiza el software BIM para comprender cómo su diseño y funcionalidades influyen en los procesos de modelado, incluyendo las características BIM incorporadas en el software. Este análisis es fundamental para explicar por qué ciertas áreas del modelado pueden beneficiarse de la ejecución de programas de asistencia.
3. **Evaluación de los potenciales de la programación paramétrica-visual** Se llevan a cabo evaluaciones para entender las capacidades de la programación paramétrica-visual y cómo esta herramienta puede abordar y ofrecer soluciones a las limitaciones o desafíos identificados en el software BIM. Esto permite justificar la elección de una interfaz o lenguaje de programación como la herramienta para el desarrollo de la propuesta.

4. **Diseño e Implementación de los Módulos** Una vez comprendidos los límites y alcances normativos de las instalaciones eléctricas, así como el funcionamiento del software BIM y sus capacidades de programación, se procede al diseño y la codificación de los módulos necesarios para lograr la propuesta planteada en el objetivo de la presente tesis. Se explican las razones detrás de las decisiones tomadas en el proceso de programación y la base normativa o constructiva que justifica estas decisiones.
5. **Ejecución y Resultados** Tras la finalización del desarrollo de la propuesta, esta se aplica en un entorno arquitectónico real proporcionado por un arquitecto proyectista. En esta etapa se describe cómo se implementa la propuesta en el software BIM y cómo influye en el proceso de modelado. Los resultados obtenidos a partir de estas aplicaciones se discuten en la sección correspondiente.
6. **Discusión de los resultados** Con base en los resultados obtenidos, se discute cómo la aplicación de la propuesta afecta al modelo arquitectónico de aplicación, considerando tanto sus alcances BIM como sus efectos temporales. También se tienen en cuenta los aspectos propios del software y de la interfaz de programación para explicar como estas influyen en los resultados.

### 1.8.1 *La Investigación*

Atendiendo a los conceptos expuestos por Nallaperumal y Krishnan (2013, p. 8), el enfoque del presente trabajo en sus diferentes aspectos son los siguientes.

- **Tipo**

El método que se propone implica un estudio **Mixto**, es decir, *Cualitativo & Cuantitativo*, ya que se evalúan las variables iniciales cualitativas y sobre ellas, los módulos propuestos se ejecutan obteniendo así un resultado de cálculo y diseño.

- **Según Nivel**

El presente trabajo de tesis ubica su nivel investigativo en el tipo **Explicativo**, ya que a través del desarrollo de la herramienta informática, implica una comprensión y explicación de la relación que se halla entre el software BIM y la programación paramétrica-visual frente a su aplicación en el diseño y cálculo de un proyecto de instalaciones eléctricas interiores.

- **Según Diseño**

El diseño propuesto para el trabajo se describe como tipo **No Experimental**.

- **Según Propósito**

El propósito el cual define la investigación, encuentra su forma en dos tipos: **Aplicada & Tecnológica**; porque el algoritmo propio de los módulos configurarán propiedad intelectual para el desarrollo de flujo de trabajo BIM ideal y la vez, al formularse el problema en base a productos informáticos existentes, se buscará mejorar la funcionalidad de estos.

Tabla 1.3

Matriz de Consistencia

PROBLEMA	OBJETIVOS	HIPÓTESIS	VARIABLES	INVESTIGACIÓN	MÉTODOLOGÍA
<b>Problema General</b>	<b>Objetivo General</b>	<b>Hipótesis General</b>	<b>Variables Independientes:</b>	<b>Tipo</b>	<b>1-<i>Revisión Normativa y Conceptual</i></b>
¿Será posible la creación de una herramienta informática basada en Programación Paramétrica-Visual aplicada a software BIM para el desarrollo de la etapa de diseño y cálculo de instalaciones eléctricas interiores?	Elaborar una herramienta informática basada en Programación Paramétrica-Visual aplicada a software BIM para el desarrollo de la etapa de diseño y cálculo de instalaciones eléctricas interiores.	La propuesta elaborada, teniendo consigo las posibilidades que ofrece la Programación Paramétrica-Visual cuando es aplicada al Software BIM, permite el desarrollo de la etapa de diseño y cálculo de instalaciones eléctricas interiores.	La programación paramétrica-visual	Mixta	
				<i>Dimensiones</i>	<b>2-<i>Análisis de las características del software BIM</i></b>
				-Compatibilidad informática	
				-Método de operación	<b>3-<i>Evaluación de los potenciales de la programación paramétrica-visual</i></b>
				<i>Dimensiones</i>	
				Nivel	
<b>Problemas Específicos</b>	<b>Objetivos Específicos</b>	<b>Hipótesis Específicas</b>			
¿Qué características tiene el software BIM para el diseño y cálculo de instalaciones eléctricas?	Determinar las características que posee el software BIM para el diseño y cálculo de instalaciones eléctricas.	Las características del software BIM para la etapa de diseño y cálculo de instalaciones eléctricas responden a la comprensión de necesidades, inyección de información necesaria y posibilidades de gestión del proyecto.	El software BIM	-Complejidad de software	<b>4-<i>Diseño e Implementación de los Módulos</i></b>
				-Características arquitectónicas premodeladas	
				-Capacidad de gestión de proyecto	
				<b>Variable dependiente:</b>	<b>Diseño</b>
¿Qué posibilidades ofrece la Programación Paramétrica-Visual dentro de la especialidad de instalaciones eléctricas interiores hechas en software BIM?	Hallar las posibilidades de la Programación Paramétrica-Visual para su aplicación en la especialidad de instalaciones eléctricas de tipo interiores.	Se hallan posibilidades dentro de la Programación Paramétrica-Visual, sin embargo, se encuentran limitaciones de compatibilidad informática con el software BIM.	La etapa de diseño y cálculo de instalaciones eléctricas interiores	<i>Dimensiones</i>	<b>5-<i>Ejecución y Resultados</i></b>
				-Facilidad de generación de modelo BIM	
				-Concordancia BIM	
¿Se podrán desarrollar los algoritmos, bloques y códigos que puedan resolver el funcionamiento de la herramienta para su aplicación en software BIM?	Desarrollar algoritmos, bloques y códigos que expliquen el funcionamiento de los módulos para su aplicación en software BIM.	El desarrollo de módulos están en firme concordancia a un proceso algorítmico en el flujo de trabajo BIM para instalaciones eléctricas interiores.		-Concordancia normativa	<b>6-<i>Discusión de los resultados</i></b>
				<b>Propósito</b>	
				Aplicada y tecnológica	
¿Se podrá observar el funcionamiento de la propuesta por medio de su aplicación en una base arquitectónica?	Aplicar la propuesta a una base arquitectónica en sus alcances de diseño y cálculo de instalaciones eléctricas interiores.	La aplicación de la propuesta demuestra una facilidad de generación del modelo de instalaciones eléctricas interiores, estando acorde a las definiciones de la metodología BIM y a la norma vigente.			

Fuente: Elaboración propia.

## CAPÍTULO II

### FUNDAMENTO TEÓRICO

#### 2.1 Introducción

Considerando la problemática expuesta en el capítulo anterior, donde se evidenció la necesidad de agilizar el proceso de diseño y cálculo de proyectos de instalaciones eléctricas, se destaca que, además de los criterios de ejecución de labores que un ingeniero electricista puede tener, en el Perú existe una cultura medianamente pobre en cuanto al estricto respeto de la normativa que regula el ámbito de desarrollo de dichas instalaciones.

Con base en lo expuesto en el párrafo anterior y en sintonía con los objetivos planteados en esta tesis, el presente capítulo se enfocará en proporcionar alcances normativos y conceptuales que se consideren pertinentes y de vital importancia para lograr el desarrollo y cumplimiento de los objetivos propuestos inicialmente.

#### 2.2 Antecedentes

##### 2.2.1 *Antecedentes Locales*

###### 2.2.1.1 «Diseño de las Instalaciones Eléctricas, Especiales aplicando tecnología BIM, Automatización y Autogeneración Híbrida, de un Hotel 5 Estrellas»

Por Noa Mayta y Vergara Ovalle (2021).

Los autores, dentro de su trabajo de Tesis para optar el título de Ingeniero Electricista, refieren el trabajo más reciente centrado a Instalaciones Eléctricas bajo la metodología BIM según el repositorio oficial de la Universidad Nacional de San Antonio Abad del Cusco. El trabajo aborda el proceso de diseño, cálculo y proyección de sistemas eléctricos y especiales de una edificación calificada como un Hotel 5 estrellas

Sin embargo, en sus conclusiones, los autores destacan que si bien utilizaron el software *Revit* de *Autodesk*, diseñado para aplicar la metodología BIM, su uso se limitó a una función complementaria en el flujo de trabajo, ya que aún se vieron obligados a trabajar previamente en un software CAD antes de migrar al entorno BIM. Argumentan que esto implicó realizar un trabajo duplicado y consideran que se necesita una base de datos sólida y un manejo experto del software para optimizar el proceso.



Si bien la primera afirmación en tanto a la base de datos existente puede ser considerada subjetiva, ya que esta puede ser proporcionada por el propio usuario a través de un minucioso trabajo de modelado de elementos BIM; la segunda afirmación demuestra que la especialidad de instalaciones eléctricas hechas en software BIM requieren una simplificación de uso o por lo menos un mejor flujo de trabajo.

## **2.2.2 Antecedentes Nacionales**

### **2.2.2.1 «Propuesta de workflow BIM/VDC para el diseño geométrico de vías de transitabilidad de la Zona Q - Huaycán Ate-Lima»**

Por Rojas Blas (2021) en la Universidad Cesar Vallejo.

El autor en su tesis de licenciatura refiere a la adopción BIM en un proyecto de infraestructura vial. Aunque el enfoque principal de su tesis está en la implementación de BIM de otra especialidad de ingeniería, se destaca que dentro del desarrollo de su proyecto, se hace mención de la herramienta *Dynamo*. Tomando esta en un nivel de desarrollo básico para su interacción con el software BIM utilizado en su caso (*Autodesk Civil 3D*). Aunque en sus conclusiones no se menciona explícitamente *Dynamo*, la herramienta se mostró útil en el desarrollo de su especialidad.

### **2.2.2.2 «Propuesta de uso de la herramienta dynamo para optimizar el tiempo en el proceso de valorización de subcontratistas de acabados en edificios multifamiliares menores de 10 pisos de NSE A y B ejecutado por una Pyme constructora en Lima Metropolitana»**

Por Chavez Guerrero y Vilchez Chung (2021) en la Universidad Peruana de Ciencias Aplicadas.

En la tesis en mención, los autores abarcan distintos campos del control de obra y valorización, incluyendo a la especialidad de instalaciones eléctricas. Todo esto en un marco de mención solamente, ya que la herramienta de *Dynamo* junto a su propuesta, configuran su utilidad para la etapa de “ejecución de obra”, llevando la operación de la herramienta *Dynamo* en algoritmos basados en una toma de control de metrados y avances de obra.

## **2.2.3 Antecedentes Internacionales**

### **2.2.3.1 «Parametric design with Visual programming in Dynamo with Revit»**

Por Nezamaldin (2019) para KTH Royal Institute of Technology in Stockholm en Suecia.

En la tesis de maestría mencionada, el autor desarrolla conceptos clave sobre lo que él denomina “diseño paramétrico” dentro de la metodología BIM. Su investigación destaca que BIM va más allá de la simple representación en tres dimensiones y que es fundamental proporcionar la mayor información posible para obtener mejores resultados en los proyectos. En este contexto, *Dynamo* se destaca como una herramienta valiosa para la automatización del proceso de diseño, lo cual es enfatizado en sus conclusiones. Además, en la parte final de su trabajo (p. 35), el autor alienta a los profesionales de ingeniería a familiarizarse más con *Dynamo* y desarrollar módulos o scripts cada vez más poderosos para la industria de la construcción.

### **2.2.3.2 «Integrated BIM and DfMA parametric and algorithmic design based collaboration for supporting client engagement within offsite construction»**

Por Bakhshi et al. (2022) para la revista científica *Automation in Construction Journal*.

El artículo citado conversa sobre el desarrollo de un algoritmo en *Dynamo* con fines de automatizar los cambios en un proyecto realizado en *Autodesk Revit* por pedido y alcances de un cliente.

Los autores destacan la necesidad de potenciar los beneficios del BIM, ya que, desde una perspectiva económica y ambiental, la metodología ha demostrado una eficacia significativa para reducir los desperdicios de materiales en la etapa de construcción.

Además, hacen hincapié en que, al igual que Nezamaldin, indican que *Dynamose* debería desarrollar en otras más disciplinas por sobre la que ellos están enfocando, que en su caso es, la mueblería y los vanos de un proyecto de edificación.

A su vez, también exponen que para tareas mucho más complejas fue necesario recurrir a la programación vía interfaz de programación *Python*.

Es de importante mención también que, los autores afirman que frente a la posibilidad de que el entorno de *Dynamo* se vea como un código de programación, este no debe ahuyentar a los profesionales ni clientes, ya que esta herramienta ofrece posibilidades a trabajar bajo una interfaz de usuario basada en nodos para su fácil uso e implementación.

### **2.2.3.3 «Diseño de modelos BIM para infraestructuras civiles con DYNAMO»**

Por Benavente Lepe (2019) para la Universidad de Sevilla en España.

En la tesis presentada por el autor, enfoca su investigación en la sección estructural del diseño de un proyecto utilizando *Dynamo*. Aunque las especialidades como instalaciones eléctricas o sanitarias no se encuentran dentro de sus alcances, sus conclusiones son relevantes, ya que destacan la interoperabilidad que se puede lograr con *Dynamo*, especialmente en relación con la compatibilidad

y manipulación de software como *Microsoft Excel*, el cual es ampliamente utilizado por profesionales del área de ingeniería para cálculos, diseño y documentación de metrados y presupuestos.

### 2.3 Marco Normativo

Como parte de sustento normativo, mismo que hace posible y respalda el correcto desarrollo de módulos que funcionen en cumplimiento a normativa peruana vigente dentro de un proyecto de instalaciones eléctricas, para el presente trabajo de tesis se tomará como la base normativa a cumplir los siguientes mencionados:

- Código Nacional de Electricidad (CNE) - Tomos de Utilización y Suministro. Norma provista por el Ministerio de Energía y Minas (MINEM).
- Reglamento Nacional de Edificaciones (RNE) - Apartados normativos para el desarrollo de Instalaciones Eléctricas y Mecánicas. Norma provista por el Ministerio de Vivienda, Construcción y Saneamiento (MVCS).
- Norma DGE - Símbolos Gráficos en Electricidad. Norma provista por la Dirección General de Electricidad del Ministerio de Energía y Minas.
- Normas Técnicas Peruanas (NTP) - Las cuales deben estar vigentes y/o actualizadas a la fecha de redacción de este trabajo. Norma provista por distintos Comités Técnicos de Normalización.

Los elementos normativos del anterior itemizado constituyen la base normativa fundamental para el logro de los objetivos planteados del presente trabajo. Aunque éstas posean sustento técnico adicional en distintas Normas Internacionales, en caso de no encontrarse base normativa nacional, se recurrirá a sus correspondientes extranjeras.

### 2.4 Marco Conceptual

- **Programación** Para Fuentes-Rosado y Moo-Medina (2017), la programación está ligada al verbo que origina su sustantivo. Programar es plasmar una solución a un problema mediante la aplicación de un lenguaje de programación que domine el usuario a cargo de la tarea. Las soluciones que este plasme pueden ser de carácter subjetivo, ya que, cada programador tendrá sus propios criterios y métodos de solución al problema en cuestión.
- **Programación Tradicional** Según McConnell (2004, pp. 7, 464), la programación a nivel tradicional se conoce también como la escritura de código fuente. Este mismo implica plasmar

instrucciones, estructuradas o no, en líneas de código bajo un lenguaje de programación. Este también demanda un manejo limpio en técnicas y habilidades para la solución de problemas para de esa forma gestionar una interfaz de programación limpia, y así esta implique una gran cantidad de líneas de código, se alcance la calidad del programa a desarrollar.

- **Programación Visual** La Programación Visual, cayendo en la redundancia, configura una forma de realizar tareas de programación bajo una interfaz de fácil visualización, la cual a través de los elementos visuales de este, se logre, en esencia, llevar un lenguaje de programación de líneas de código a medios de rápida evaluación de resultados, llevándonos así a tener a la actividad de programación con una menor dificultad para el usuario. (Burnett y McIntyre, 1995, p. 11)
- **Módulo** McConnell (2004, p. 100) también define lo que significa un módulo en el ámbito de la programación. Este puede comprender desde una serie de instrucciones simples que pueden o no depender de una variable, hasta un conjunto de pequeños módulos que como un todo funcionan como un programa en sí o una herramienta de desarrollo para otros programas.
- **Parametrización** Dentro del campo del BIM, según Bianchini y Nicastro (2018, p. 3) la definición más acertada sobre la parametrización consiste en brindar información concerniente a la representación geométrica de un elemento constructivo, conjunto de estos o hasta un proyecto completo; convirtiendo esa característica en parámetro BIM.
- **Software BIM** Para Xiaolei (2018), se toma al Software BIM como la herramienta necesaria e inherente para el correcto desarrollo de la metodología BIM, la misma herramienta puede ofrecer mejores resultados cuando para su uso se destinan recursos computacionales de hardware mayores a un trabajo promedio bajo CAD.
- **Workflow** También denominado en habla hispana como Flujo de trabajo, término utilizando en el ámbito del BIM el cual se define como un procedimiento que posibilita la integración de los profesionales, técnicos y trabajadores que son parte del proceso constructivo. La ventaja de su propuesta y utilización es mejorar competencias de planificación de tiempos, flexibilidad ante nuevos eventos, manejo de información y la toma de decisiones (Egea et al. 2015, p. 21).
- **Instalación Eléctrica Interior** En primer lugar, se establece a la Instalación Eléctrica de Tipo Interior como el conjunto de alambrado y accesorios que se ubican en dentro de un terreno, predio o edificación; desde un punto inicial de suministro de energía (pudiendo ser una empresa concesionaria u otra entidad) hasta los puntos donde se haga uso conexionado del equipo dependiente de esa misma energía eléctrica. (Ministerio de Energía y Minas, 2006)

- **El Proyecto de Instalaciones Eléctricas Interiores** Actualmente, dentro del territorio peruano y por ende, la normativa que rige dentro de los límites de este; especifican con gran precisión aspectos varios de una Instalación Eléctrica y a su vez un Proyecto completo de esta misma. De forma resaltante y como puntos más importantes se encuentran los alcances que ofrecen el Código Nacional de Electricidad–Utilización y el Reglamento Nacional de Edificaciones.

Así mismo, tomando en cuenta su naturaleza de proyecto y su respectiva documentación, según la *Norma EM.010* (Actualizada al año 2019) por el Reglamento Nacional de Edificaciones (Ministerio de Vivienda, Construcción y Saneamiento, 2006, pp. 6-8); un proyecto de instalaciones eléctricas debe considerar documentos varios (dependiendo del tipo de edificación); el certificado de factibilidad de servicios, una memoria descriptiva con los alcances criteriosos que el proyectista utilizó para el diseño; también se debe acompañar una memoria de cálculo que justifique y sustente el trabajo de ingeniería dentro del diseño. La representación visual del proyecto debe estar acompañado además por los planos correspondientes plasmados en tamaños de papel y consideraciones técnicas avalados por las Normas Técnicas Peruanas (NTP) correspondientes.

## 2.5 Marco Teórico

### 2.5.1 *Diseño y Cálculo de Instalaciones Eléctricas Interiores*

Aunque en el Perú y en el Mundo, la tarea de proyectar instalaciones eléctricas es llevada a cabo con gran mayoría guiándose por los criterios que el ingeniero proyectista toma para su desarrollo, no obstante, y de carácter no obligatorio, Dueñas Ponce de Leon (2018) propone una metodología de diseño y cálculo basada en la planificación proyectual, cuya organización planteada de forma secuencial permita llegar a los objetivos de forma controlada.

A su vez, Dueñas Ponce de Leon recomienda cuatro aspectos puntuales en la elaboración de un proyecto de instalaciones eléctricas, estos son:

- Conocimiento Exhaustivo del Programa
- Estudio de Cargas Eléctricas
- Anteproyecto de las Instalaciones
- Elaboración Preliminar del proyecto.

Dentro de la Etapa de Diseño, el Conocimiento Exhaustivo del Programa implicará parte vital para una aplicación metodológica del BIM, ya que se toman primeros aspectos de coordinación entre disciplinas, siendo la base arquitectónica el punto más importante, la cual es provista por el arquitecto proyectista; agregando a esta otras adicionales como lo son la especialidad estructural y sanitaria.

### 2.5.1.1 La importancia de la Base Arquitectónica

Dueñas Ponce de Leon (2018, p. 17) indica que dentro del conocimiento del programa en el campo arquitectónico se deben considerar tres puntos importantes.

- La Información General – Contempla posiciones de techos, alturas y cotas de vanos e información de paredes, techos y visualización de cortes.
- Aspectos Decorativos – Formas de Ambientes, Colores, Detalles arquitectónicos.
- Ambientes de Uso Especial – Información del tipo de edificación del proyecto.

Tomar en cuenta el itemizado anterior implica un mayor conocimiento de los objetivos esperados para el proyecto en cuestiones de Diseño, mientras que el Cálculo provendrá de los aspectos técnicos de la carga instalada en base a la base Arquitectónica.

También conocer este alcance arquitectónica nos provee un camino para enlazar estudios lumínicos junto a las expectativas decorativas y de diseño de interiores. Aunque este último tiene carácter de variar según el especialista a cargo de la decoración, el trabajo de ingeniería a partir de los estudios de iluminación pueden ser sustentados incluso bajo norma base como el Anexo de la *Resolución Ministerial N° 083-2019-VIVIENDA* incluido en el Reglamento Nacional de Edificaciones.

## 2.5.2 Modelado de Información de Construcción o BIM

### 2.5.2.1 Breve Antecedente Histórico

El término *BIM*, siglas del término anglosajón *Building Information Modelling* y conocidos en habla hispana como **Modelado de Información de Construcción**; no es un concepto el cual sea relativamente nuevo a las últimas dos décadas, de hecho según Popov (2016, p. 7) el término BIM fue propuesto por primera vez en el año 1975 por Chuck Eastman, profesor norteamericano del Instituto Tecnológico de Georgia (*Georgia Institute of Technology*).

En el artículo publicado por González Marquez et al. (2014, pp. 4-5) en la revista *Spanish Journal of Building Information Modelling*, se describe también que el BIM evolucionó a partir de las posibilidades del dibujo asistido por computador en *2D* y posterior desarrollo *3D*, sin embargo

dadas las limitaciones tecnológicas de la época, fue recién para los inicios del siglo XXI cuando se empezaron a desarrollar pequeños softwares que intentaban emular la idea fundamental del BIM.

Finalmente, González Marquez et al. también dan a conocer que es para el año 2011, y años posteriores cercanos, que gobiernos como el Reino Unido y otros europeos elaboran su respectivo Plan BIM y su aplicación sobre todo en proyectos públicos.

#### **2.5.2.2 BIM**

Aunque en el mundo han habido infinidad de instituciones y profesionales que han hecho definiciones respecto al término BIM, para el presente trabajo, teniendo en cuenta el enfoque nacional normativo expuesto en uno de los objetivos, se tomará la siguiente definición: *“una metodología de trabajo colaborativo para la gestión de la información de una inversión pública, [...] para facilitar la programación multianual, formulación, diseño, construcción, operación y mantenimiento de la infraestructura pública, asegurando una base confiable para la toma de decisiones.”* (Ministerio de Economía y Finanzas, 2021).

Se comprende así que el BIM enfoca no solamente la etapa de diseño, como es comúnmente entendido, ya que entre profesionales existe la falacia que el BIM es una forma de representar diseño arquitectónico o de ingeniería de forma tridimensional sin mucha mayor información que gestionar.

Así mismo, teniendo en cuenta que la definición del Ministerio de Economía y Finanzas indica las distintas dimensiones que contiene el término BIM, por solamente la etapa de diseño, estas dimensiones se indican en la siguiente sección.

#### **2.5.2.3 Dimensiones del BIM**

Dentro de la metodología, ya que se consideran en varios enfoques y etapas del proceso constructivo con la finalidad de integrarlos a uno solo el cual ofrezca potencial para su presente y futura gestión, se definen siete (7) dimensiones en las cuales reposa toda la metodología de forma cronológica y que entre ellas se relacionan. Estas son las que se enumeran en la Tabla 2.1.

#### **2.5.2.4 El LOD**

Level of Development (LOD) o en español “Nivel de Desarrollo”, es un acrónimo que se puede llegar a confundir fácilmente con un significado también válido para el acrónimo en BIM, *Level of Detail* (Nivel de Detalle).

Según las definiciones expuestas por Madrid (2014) sobre la terminología aplicada al BIM, ambas son consideraciones válidas, sin embargo definen dos cosas distintas. Por un lado cuando se habla de detalle implica en la forma en cómo está representado gráficamente un elemento. El dibujo

**Tabla 2.1***Dimensiones del BIM*

Dimensión	Nombre	Descripción
1D	La Idea	¿Cómo será?
2D	El Boceto	Definición de la carga energética
3D	La visualización	Representar las dimensiones anteriores
4D	Los tiempos	Programar obra y la Ruta Crítica
5D	El Costo	Estimar costo de construcción y gestión
6D	El Comportamiento	Evaluar Sustentabilidad y Ahorro Energético
7D	La Operación	Gestionar el producto final

*Fuente:* Adaptado a partir de Gonzales Villamil y Lesmes Fabián (2017, pp. 75-76)

asistido por computador ofrece esta posibilidad ya que a través del dibujo vectorial se puede lograr imprimir un nivel de detalle deseado, la figura 2.1 ofrece una referencia sobre este concepto.

**Figura 2.1***LOD como Nivel de Detalle*

*Fuente:* Elaboración propia

Visto de forma intuitiva y comprensible, mientras más lograda sea la representación gráfica y se pueda distinguir mejor las características del elemento se puede concluir que el objeto ha logrado ser representado con un Nivel de Detalle mayor.

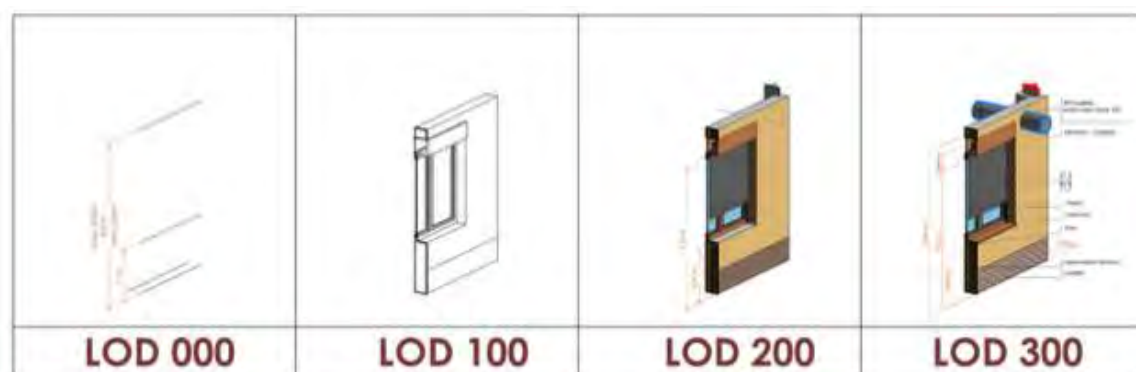
Ahora, por concepto adyacente, el Nivel de Desarrollo está enfocando al BIM por sobre el dibujo asistido por computador *CAD*, ya que, este se implica que un objeto que se relaciona con el proyecto logra mayores niveles de información en relación al proyecto en mención.

Como se puede observar en la Figura 2.2, el nivel de desarrollo puede ser medido bajo una numeración, esta numeración es comúnmente subjetiva y depende de los requisitos de LOD que establezcan instituciones gubernamentales, instituciones técnicas privadas o hasta profesionales que desarrollan o supervisan el proyecto; sin embargo Madrid (2014) ofrece unos referenciales (Ver Tabla 2.2).



**Figura 2.2**

*Ejemplo del Nivel de Desarrollo de un Objeto*



*Fuente:* Tomado de Madrid (2014, Figura 13)

**Tabla 2.2**

*LOD en el BIM*

LOD	Descripción	LOD	Descripción
<i>LOD100</i>	Representación básica bajo símbolo	<i>LOD400</i>	Se incluye información sobre la fabricación y programación de tiempos.
<i>LOD200</i>	Información aproximada de posición, dimensión, cantidad y hasta costo.	<i>LOD500</i>	Elemento en referencia cómo está quedando en obra ( <i>As Built</i> ). Puede incluso ser independiente a niveles LOD inferiores.
<i>LOD300</i>	Forma precisa, cantidades, posición y montaje respecto al proyecto y sistema constructivo.	<i>LODX000</i>	Ofrecer posibilidades de gestión futura en cuanto a su renovación y hasta su desuso o demolición.
<i>LOD350</i>	Desarrollo independiente de las especialidades y su respectiva coordinación.		

*Fuente:* Adaptado a partir de Madrid (2014, pp. 43-47)

### **2.5.3 Software BIM aplicado de la Ingeniería Eléctrica**

Para la aplicación de la metodología BIM dentro del campo de la Ingeniería Eléctrica, es necesario el trabajo conjunto con el área informativa de un proyecto, es decir, hay una necesidad de abordar aspectos y propiedades o parámetros de los elementos que sean parte de la etapa de proyección de las instalaciones eléctricas de un proyecto, todos estos deben estar representadas en una plataforma virtual la cual pueda procesar el el diseño propuesto por el ingeniero electricista.

El software BIM se hace presente de esta manera y es similar a lo que sucedía con las herramientas computaciones CAD que daban posibilidades infinitas en cuanto a la representación gráfica en ingeniería.

En tal caso, para la segunda década del siglo XXI, existe un número considerable de empresas a lo largo del mundo que se dedican a este fin, desarrollando herramientas que implementen el BIM en sus distintas dimensiones y bajo normativa tipo modificable o que este sustentada y diseñada para algún territorio nacional o continental específico. Algunas de estas son:

- Autodesk.
- Graphisoft.
- Bentley.
- Grupo NEMETSCHEK

### **2.5.3.1 Importancia de la elección del software BIM**

En el escenario donde en el mercado se tiene una gran cantidad de oferta en cuanto a Software BIM, llegado el momento de realizar tareas de diseño y cálculo para un proyecto de instalaciones eléctricas interiores, se deben tener consideraciones frente a este ya que el software BIM se presenta como la interfaz visual que deba permitir la interacción entre el usuario ingeniero electricista y la metodología BIM que desea aplicar a su proyecto.

La importancia de la elección del software para el desarrollo no solamente radica en las limitaciones informáticas e de incompatibilidad que se podrían tener entre uno y otro producto, ya que hoy en día, la interoperatividad entre softwares, si bien no es perfecta, es posible. Por lo tanto, tal relevancia se enfoca más bien en la capacidad del software para ofrecer un diseño y operación que tenga la mayor relación con lo que el ingeniero encargado busca para su proyecto.

En caso de la adecuación de las características BIM al software, cada uno de estos programas tiene sus propias maneras de abarcarlas; en tal caso, se conoce que no en todo el globo terráqueo se mantienen mismos sistemas constructivos y por ende, mismas formas de desarrollar instalaciones eléctricas. Este aspecto se da más que todo por las diferentes normas que rigen en territorios nacionales y extranjeros; aunque la mayoría de estos tienen alcances por normativas internacionales, dentro de un sustento de cálculos, la norma nacional tiene mayores relevancias.

Finalmente, un factor determinante en la elección de software es la capacidad de comprender a la especialidad como una gama de aspectos técnicos más allá de un modelo 3D que contenga solamente información visual. Hoy en día, bastantes son los desarrolladores de software BIM que enfocan las funcionalidades de su producto a las características visuales de la especialidad de arquitectura y acabados; siendo más difundidos conjuntos de configuración dirigidos al renderizado de imágenes como parte de un proceso de venta y propaganda para un proyecto, teniendo más dejados de lado configuraciones técnicas para el desarrollo de ingeniería de instalaciones eléctricas.

#### 2.5.4 Programación Paramétrica-Visual

Aunque la programación Visual no es un término relativamente nuevo como expuso Burnett y McIntyre (1995), al igual que la programación Digital, aún al siglo XXI existen bastantes enfoques con el cual poder conceptualizar o brindar una idea sobre esta.

En tanto se pueda comprender que “programar” es traducir un algoritmo a un lenguaje de programación el cual pueda aplicarse a cierta tarea, bastaría con retrotraer esa misma idea al flujo de trabajo convencional en un proyecto de edificación o hasta de infraestructura.

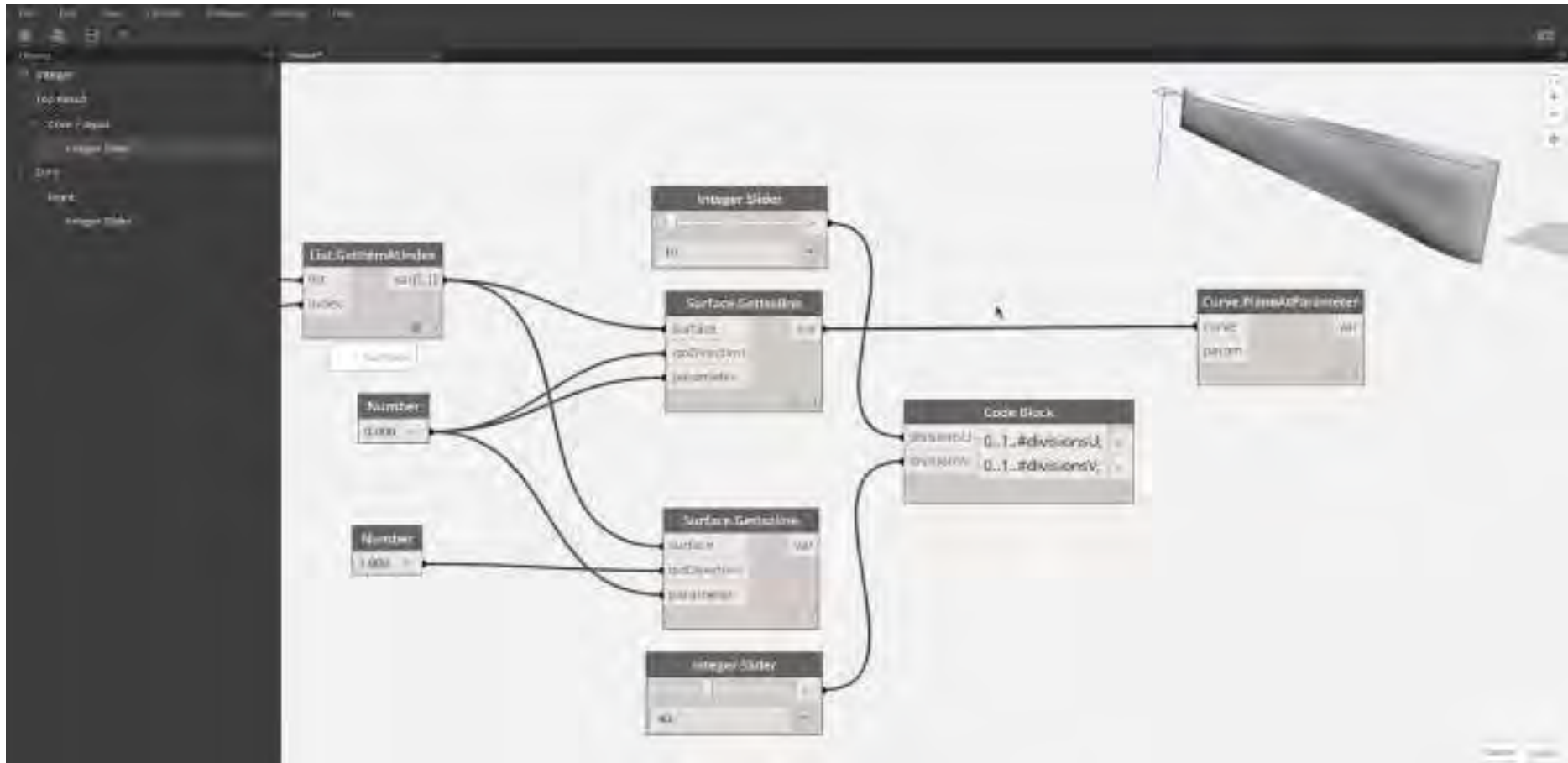
Llevando la definición inicial, junto a la naturaleza Paramétrica de los modelos BIM, el conjunto *Programación Paramétrica-Visual* implicará el relacionar una interfaz de Programación Visual a un modelo BIM el cual se pueda manipular dentro de sus alcances paramétricos.

Ahora bien, considerando que el profesional realiza durante el proceso de diseño y cálculo es seguir una serie de pasos el cuál le permita desarrollar sus ideas y criterios plasmándolos en un documento técnico conocido como “plano”, esa serie de pasos se puede conceptualizar de manera muy directa con lo que es algoritmo; siguiendo esta secuencia, podremos ampliar la idea de programación digital, a desarrollar algoritmos bajo un código o lenguaje de programación bajo no solamente bajo líneas de código sino también elementos visuales de diseño, por fin estaríamos encontrando una forma de concebir la programación paramétrica-visual.

También hay argumentos para considerar que en el campo del diseño, este tipo de programación lleva cierta ventaja de uso. “*El aspecto visual de la programación permite reducir la barrera en relación con la entrada y, con frecuencia, ofrece mayor claridad a los diseñadores*” (Dynamo Primer, 2019).

**Figura 2.3**

*Interfaz de Programación Paramétrica Visual*



*Fuente:* Elaboración propia

#### 2.5.4.1 Diferencias: Programación Tradicional & Programación Paramétrica–Visual

Es probable que cuando se comienza con el verbo se programar solamente se tenga en imagen mental el aprendizaje de un lenguaje diferente entre distintas interfaces junto a líneas de código numeradas, teniendo que llegar a la comprensión de una sintaxis para la realización de tareas en específico. Pues es esta misma idea la que rodea en general a lo que precisamente se conoce como *Programación Tradicional* o textual.

Considerada de esa forma, el trabajo en un entorno tradicional se vuelve un proceso de comunicación entre usuario programador y la interfaz, en el que si la complejidad del programa a desarrollarse se eleva, el planteamiento de algoritmos lo hará también y como consecuencia directa en forma cuantitativa, el número de líneas de código también será totalmente mayor.

En lo que concierne a los entornos visuales de programación, según Shu (1986, p. 33), hay muchas ramificaciones en las que es posible ubicar a los métodos de programación visual; pero en virtud a la implementación y uso de entornos visuales, sean cuales fuese la aplicación, se explica que transformar la programación tradicional o “formal” a un entorno de imágenes puede traer simplificaciones de lenguaje, una gran cantidad de elementos gráficos como parte negativa y un alcance de programación mucho más limitado. Aquí entra en relevancia la importancia de que se pueda ampliar el entorno visual a uno de tipo manipulable o ahora llamado de tipo “paramétrico”. Shu explica de forma comparativa que frente a la limitación de un entorno visual solo gráfico, tener un enfoque manipulable probablemente haga que el lenguaje de programación ya no se convierta en algo tan sencillo pero tampoco tan complejo como en la programación tradicional; a su vez, la extensión visual de la cantidad de elementos gráficos será mucho menor pero logrando un alcance programable mucho mayor incluso que la forma tradicional de programar. Esta comparativa se puede apreciar en forma de un manejo de ejes gráficos en la Figura 2.4.

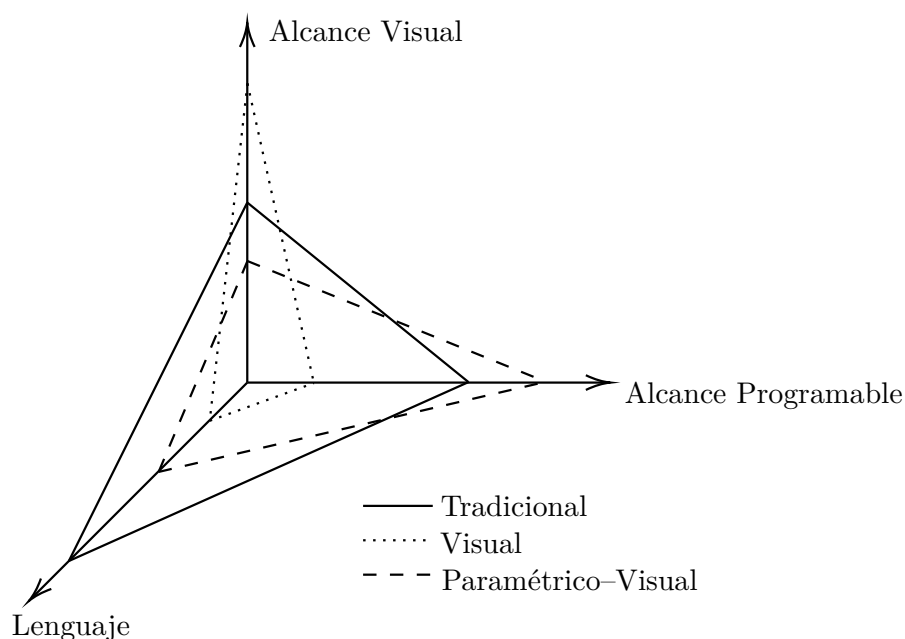
Vistas las diferencias en una forma de pilares importantes para la tarea de programación, esta comparativa puede ser ampliada en aspectos muchos más descriptivos en cuanto a características computacionales que deben considerarse al momento de tomar a la programación paramétrica–visual; sobre todo al considerar que actualmente nos encontramos en auge de tecnologías y la comparativa hecha por Shu, aunque es válida, fue realizada tomando en cuenta el contexto de la época (1986).

Esto quiere decir que, hoy en día, se han de considerar también características relacionadas al tamaño de archivos, usos comunes, impacto económico. En la Tabla 2.3 el autor de la presente tesis expone una ampliación de las diferencias entre una programación tradicional y la programación paramétrica–visual.

En primer lugar, la diferencia más significativa a primera vista es la interfaz que se puede manejar, teniendo la programación tradicional las ya conocidas líneas numeradas mientras que su

**Figura 2.4**

*Comparativa de los entornos tradicional, visual y visual-paramétrico de programación*



*Fuente:* Adaptado a partir de Shu (1986).

contraparte posee comúnmente bloques con líneas de enlace o geometrías que permitan la tarea programable (Ver Figura 2.5).

**Figura 2.5**

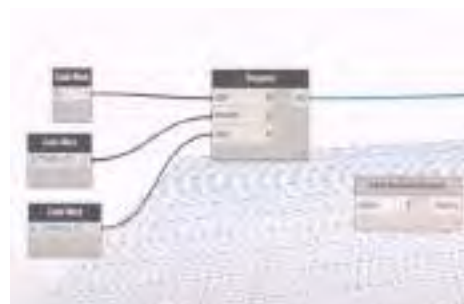
*Interfaz tradicional vs. visual*

```
public class CalculoCuadrado {
    public static void Main(String[] args) {
        SinMetodoMain main = new SinMetodoMain();
        int resultado = main.CalculaCuadrado(15);
        System.out.println(resultado);

        Mensaje m = new Mensaje();
        m.Mostramensaje();

        System.out.println("FIN DEL PROGRAMA");
        System.out.println();
    }
}
```

(a) Líneas de código.



(b) Bloques y enlaces.

*Fuente:* Elaboración propia

Seguidamente, dentro de los propósitos en los que se usan, es sabido que el entorno tradicional tiene aplicaciones infinitas, sin embargo, su uso dedicado y por la complejidad del mismo, su propósito ha sido más dirigido a fines profesionales. En parte de la programación paramétrica-visual, se han hallado propósitos profesionales y a la vez educativos ya que dentro de planes de estudio dirigido a niños y adolescentes, se propone software de tipo entorno visual para la enseñanza más sencilla y eficaz de programación a estos (Ver Figura 2.6).

**Tabla 2.3***Comparativa Programación tradicional vs. Programación paramétrica-visual.*

	<b>Programación Tradicional</b>	<b>Programación Paramétrica-Visual</b>
<i>Interfaz</i>	Lineas numeradas	Bloques, nodos u otros elementos visuales.
<i>Propósitos</i>	Profesionales	Educativos y Profesionales.
<i>Sintaxis</i>	Aprendizaje obligatorio.	Aprendizaje opcional.
<i>Tamaño informático</i>	Ligero	Muy pesado
<i>Diagnóstico de errores</i>	Ubicación de línea de código.	Ubicación de bloque deficiente.
<i>Complejidad del diagnóstico</i>	Tedioso.	Visual respecto al flujo de datos.
<i>Impacto económico</i>	Alto al desarrollo de software.	Menor al desarrollo de software.
<i>Impacto de la interfaz</i>	Alto cuando lleva gran número de líneas.	Alto y tedioso cuando lleva un gran número de bloques.

*Fuente:* Elaboración propia

En tanto a la sintaxis, todo lenguaje de programación tradicional demanda en si mismo conocer de qué manera llegan los datos a los comandos a usarse en el orden debido; es este factor el cual en la mayoría de casos provoca que la programación tenga una curva de aprendizaje muy compleja. Esta misma sintaxis tiene carácter opcional en enfoques visuales ya que esta permite el desarrollo de elementos personalizados a tareas en específico, conocer la sintaxis apoya al potencial programable; sin embargo, no configura en sí una obligatoriedad para iniciar tareas de programado.

El tamaño informático, en tiempos donde se poseen tecnologías de nubes virtuales alojadas en internet, ha dejado de ser una preocupación grave, no obstante, con fines comparativos, un programa hecho en un entorno tradicional será mucho más ligero que uno desarrollado en entornos visuales; esto es debido a que en ejecución de lineas de código numeradas, el peso informático implica a procesar textos solamente, mientras que en usos visuales, la representación de enlaces y bloques invoca una mayor necesidad de espacio en una computadora.

En cuanto a la toma de errores en su alcance de diagnóstico, la programación de forma tradicional requiere no solamente ubicar la línea de error, sino que también hace que sea necesario conocer las sintaxis necesarias para poder identificar y levantar el error; calificando así su forma

**Figura 2.6**

*Interfaz de Scratch como entorno visual de programación con fines educativos.*



*Fuente:* Elaboración propia

de tedioso el trabajo de diagnóstico. La programación paramétrica–visual por su parte, al no ser obligatorio el aprendizaje de una sintaxis, el diagnóstico de errores se hace más cómodo ya que las interfaces resaltan los elementos donde se encuentran errores e incluso desde dónde se está arrastrando algún otro.

Se llega así a considerar también que cuando es necesario el desarrollo de programas, frente a entornos tradicionales es posible que se requiera a un ingeniero de la especialidad o personal altamente calificado para este desarrollo; mientras que el entorno visual, al tener la capacidad de que puede ser enseñado incluso a niños y adolescentes, lleva a que la tarea de desarrollo de programas lo pueda hacer el mismo usuario interesado, llevando consigo de esta manera un alivio económico en comparación a una propuesta tradicional.

Sin embargo, entre métodos tradicionales y visuales, se encuentra un punto de convergencia en donde es requerido tomar en cuenta cuando se pretende un desarrollo de programas muy complejos. Visto de esta forma, el primero llevará una gran cantidad de líneas, trayendo consigo un agregado a la dificultad si es que surgiese algún error; mientras que dentro de la misma complejidad, si el programa se llevase a la representación de bloques y enlaces, el número de estos crecería de igual forma, provocando más que todo un copiamiento visual de la interfaz, pudiendo generar problemas de orden (Ver Figura 2.7).



**Figura 2.7**

*Copamiento visual en diferentes entornos de programación*



(a) *Gran número de líneas para un programa.*



(b) *Gran número de bloques y enlaces para un programa.*

*Fuente:* Elaboración propia

### 2.5.4.2 Interfaces de Programación

Del mismo modo que sucede con lenguajes de programación tradicionales, en la actualidad, para la programación paramétrica-visual existe un grandísimo número de lenguajes que se pueden utilizar para distintos fines, mas son muy contados lo que están dirigidos a aplicarse a software enfocado a la aplicación de la metodología BIM.

La existencia de interfaces visuales con objetivo de desarrollo BIM destaca por encima de sus funcionalidades la compatibilidad que le ofrece el software BIM a estos. A su vez, el hecho de que la mayoría de estas interfaces hayan sido desarrolladas en la última década, ofrece posibilidad abierta a que haya comunicación con repositorios web que contengan recursos adicionales Noves (2015, pp. 21-25). En los siguientes puntos se evalúan algunos de las interfaces que logran ajustarse a las metodologías propuestas por desarrolladores de software BIM.

**2.5.4.2.1 Grasshopper 3D** Grasshopper 3D es una interfaz de programación visual dedicada al modelado 3D, cuyo funcionamiento se deriva a la recepción de datos y transformarlos en elementos geométricos. Una limitación expuesta por Kirschner (2015, p. 42) sobre el lenguaje de Grasshopper es la inexistencia de procesos recursivos como normalmente se pueden encontrar con mucha facilidad en lenguajes de programación tradicionales.

#### Figura 2.8

*Grasshopper 3D*



*Fuente:* Icono de software tomado de Grasshopper 3D (2022)

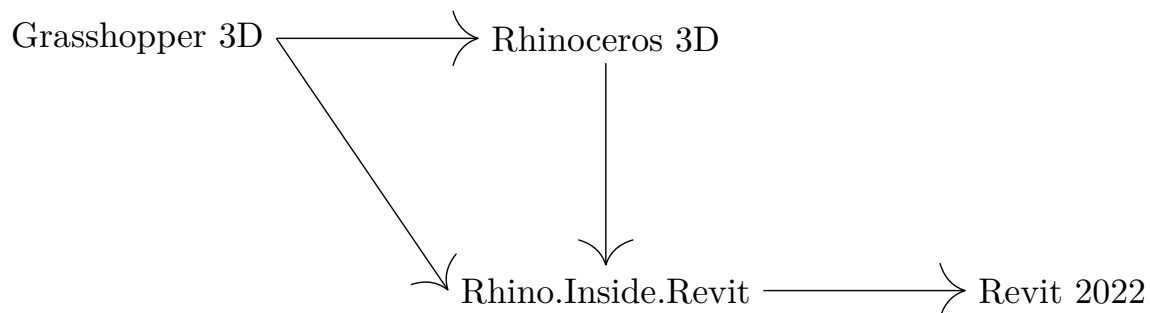
A su vez, otra limitación hallada es que según el portal web de Grasshopper 3D<sup>1</sup>, se señala que a la fecha de elaboración de esta tesis, la interfaz de programación visual solamente posee compatibilidad con el software de modelado 3D llamado *Rhinoceros 3D* desarrollada por la empresa estadounidense *Robert McNeel & Associates*.

El alcance BIM que se le puede dar a Grasshopper como una potencial herramienta a usarse es a través de aplicación tercerizada (Ver Figura 2.9) también desarrollada por *Robert McNeel & Associates*.

<sup>1</sup> <https://www.grasshopper3d.com/>

**Figura 2.9**

*Interoperatividad entre Grasshopper 3D y Revit 2022*

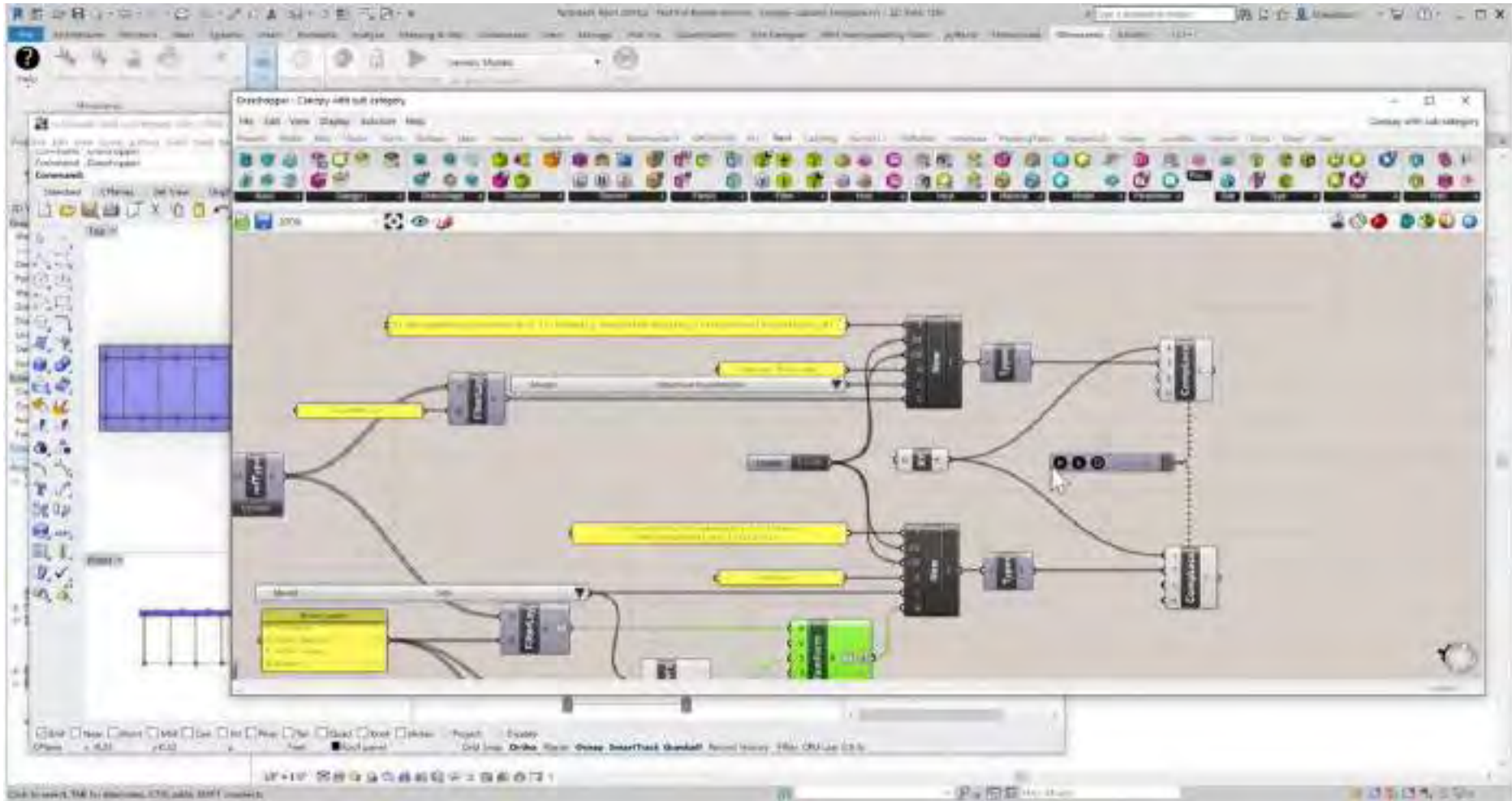


*Fuente:* Elaboración propia

Aunque el uso de adicionales o también llamados *plugins* dentro del desarrollo proyectual dentro de Revit es más común en la actualidad; en la aplicación de *Rhino.Inside.Revit* requiere el uso paralelo de tres softwares; Grasshopper como interfaz de programación, Rhinoceros como mediador de modelado 3D y finalmente Revit 2022 funcionando con *Rhino.Inside.Revit* (Ver Figura 2.10). A niveles informáticos este aspecto podría traer inconvenientes ya que los recursos computacionales a usarse en su aplicación se elevarían a comparación de una compatibilidad directa.

Figura 2.10

Uso paralelo de Rhinoceros, Grasshopper y Revit



Fuente: Interacción de software tomada de Grasshopper 3D (2022)

**2.5.4.2.2 Python** A pesar de que es el enfoque en cuestión a la interfaz de programación nos deriva a evaluar los lenguajes de programación visual, es de importante mención en el presente capítulo destacar la operatividad que posee el lenguaje de programación Python.

**Figura 2.11**

*Python Software*



*Fuente:* Icono de software tomado de Python Software Foundation (2022)

Python se aleja del enfoque visual y regresa al tratamiento tradicional de programar en interfaces textuales con líneas numeradas. La importancia de su mención se debe a que esta interfaz de programación posee una compatibilidad directa con Revit 2022, ya que este software BIM se abre a la manipulación de su código, pero esto claro demanda tener un conocimiento pleno de la sintaxis interna que maneja Revit 2022. Pero sin cabida a inconvenientes con el libre acceso a la información, actualmente en internet se encuentra el sitio web llamado *Revit API Docs*<sup>2</sup> el cual provee la totalidad de las sintaxis sin costo alguno.

El potencial de uso de esta herramienta como alternativa se debe a que, al ser un medio de programación tradicional, se pueden implementar toda clase de técnicas programables siendo las más relevantes el uso de repositorios web y técnicas iterativas y recursivas (aspecto último que por ejemplo, Grasshopper no posee).

**2.5.4.2.3 Dynamo** Al igual que Grasshopper 3D, Dynamo también es una interfaz de programación paramétrica-visual, sin embargo, recaba su principal diferencia en que este se encuentra más dirigido a complementar a la metodología BIM y menos hacia únicamente el modelado 3D; ya que según su sitio web oficial<sup>3</sup> su gama de compatibilidad abarca a software BIM, aparte de Revit 2022, con otros muchos más que fueron también desarrollados por *Autodesk* y que enfocan hacia el BIM en sus diferentes especialidades.

Dynamo al poseer una compatibilidad de tipo directa no demanda de ninguna aplicación tercera o *plugin* que haga de intermediario entre su interfaz y Revit; permitiendo así una ejecución eficiente en cuestiones de recursos computacionales. Es de importante mención también que, a dife-

<sup>2</sup> <https://www.revitapidocs.com/>

<sup>3</sup> <https://dynamobim.org/>

**Figura 2.12***Dynamo Software*

*Fuente:* Icono de software tomado de Dynamo BIM (2022)

rencia de otras interfaces, desde el año 2019 en los productos *Autodesk* que ofrece compatibilidad, su instalación es automática ya que el software BIM base viene junto a la interfaz de programación (Dynamo) y su ejecutable como aplicación (Dynamo Player). Esta instalación automática incluso se da para las versiones estudiantiles y académicas del software BIM, permitiendo de esta manera la investigación de sus potenciales.

## CAPÍTULO III

### ANÁLISIS DE CARACTERÍSTICAS DE SOFTWARE BIM PARA LA ETAPA DE DISEÑO & CÁLCULO DE INSTALACIONES ELÉCTRICAS

#### 3.1 Introducción

Dentro del contexto del Building Information Modeling (BIM) y sus implicaciones teóricas en la metodología de trabajo, es crucial realizar un análisis detallado del software que permita su aplicación en el proceso de diseño y cálculo de instalaciones eléctricas. Para lograr esto, es necesario comprender las consideraciones y criterios que emplea un ingeniero electricista durante el flujo de trabajo al encargarse del diseño y cálculo de las instalaciones eléctricas interiores en un proyecto específico.

Además, es de gran importancia identificar las principales características BIM que estarían involucradas en el trabajo en un software específico. El objetivo es comprender la naturaleza paramétrica y manipulable de dichas características para posibilitar la programación informática que se pueda aplicar al diseño y cálculo de las instalaciones eléctricas interiores.

En el presente capítulo, se desarrollará un análisis exhaustivo que permita identificar las características fundamentales del software BIM. A través de este análisis, se podrá comprender la naturaleza paramétrica y manipulable de las herramientas, lo que posibilitará la programación informática necesaria para la aplicación efectiva en el diseño y cálculo de las instalaciones eléctricas interiores.

#### 3.2 Instalaciones Eléctricas

En el desarrollo del proyecto de instalaciones eléctricas interiores, que constituye una parte vital en un proyecto de edificación, el ingeniero electricista tiene un amplio margen para asimilar la naturaleza del proyecto. Esto implica el uso de sus propios algoritmos para llevar a cabo el proceso completo de diseño, cálculo y documentación del proyecto. Esta sección amplía el enfoque académico de ese “algoritmo”, que puede pasar de ser una concepción en la mente del profesional a una herramienta informática programable para su posterior ejecución.

### **3.2.1 Consideraciones de la Especialidad**

Comparado con otra especialidad de diseño, como la Arquitectura, las instalaciones eléctricas tienen un proceso creativo distinto y se centran más en el estudio de necesidades y aspectos técnicos del proyecto. Sin embargo, el trabajo de ingeniería se despliega al estudiar y proponer soluciones que se adapten a la arquitectura de la edificación donde se implementarán las instalaciones eléctricas. Es esencial que el ingeniero electricista se mantenga en total cumplimiento de las normativas vigentes, siguiendo lo que estas dictaminan, ordenan o prohíben.

A partir de esta primera aproximación a las consideraciones de la especialidad, surge la idea de lo que se debe tener en cuenta sobre el software BIM que se utilizará para desarrollar la metodología. No obstante, el proceso de un proyecto de instalaciones eléctricas puede abarcar mucho más que estas consideraciones iniciales mencionadas anteriormente, y puede incluir criterios específicos del profesional. Como este trabajo tiene un enfoque académico, los criterios que se expondrán en la siguiente sección no deben considerarse como imperativos en el trabajo y ejercicio profesional. Se presentan a modo de implementarse a nivel de código de programación, pero también se toman en cuenta diversas fuentes y autores de la especialidad.

### **3.2.2 Criterios para el Diseño y Cálculo**

Es importante tener en cuenta que los criterios para desarrollar un proyecto de instalaciones eléctricas provienen principalmente del profesional, su formación y experiencia. Estos criterios no pueden establecerse como postulados académicos universales, ya que varían según el contexto y las necesidades específicas de cada proyecto. Sin embargo, dentro de la formación impartida en la Universidad Nacional de San Antonio Abad del Cusco y la Escuela Profesional de Ingeniería Eléctrica, Dueñas Ponce de Leon (2018, pp. 27-29) expone tres criterios importantes que se deben tomar en cuenta:

#### **3.2.2.1 Seguridad**

El criterio de seguridad es fundamental en el diseño de instalaciones eléctricas. Se debe garantizar la seguridad de los usuarios, incluidos seres humanos, animales y plantas. Además, se debe asegurar la protección del equipo eléctrico y de la carga que se conectará a la instalación. La normativa vigente en el Perú juega un papel importante en el cumplimiento de este criterio.



### 3.2.2.2 Flexibilidad

El criterio de flexibilidad abarca aspectos técnicos y de ingeniería en la instalación eléctrica. Se evalúa el sistema de utilización desde las acometidas hasta los últimos circuitos derivados y otras instalaciones relacionadas con el proyecto. La evaluación busca determinar la posibilidad de que la instalación ofrezca continuidad en el suministro de energía, regulación de tensión eléctrica y espacio para ampliaciones futuras sin necesidad de demoler o desechar el trabajo realizado.

### 3.2.2.3 Funcionamiento

El criterio de funcionamiento se logra al cumplir los dos criterios anteriores y asegura un correcto desempeño de la instalación en conjunto. Para lograrlo, se deben tener en cuenta las siguientes particularidades por proyecto:

**3.2.2.3.1 Factores y su Aplicación** Es fundamental emplear factores durante el proceso de cálculo y replanteamiento del diseño para evitar errores de sobre o sub dimensionamiento de los elementos de la instalación.

**3.2.2.3.2 Caídas de Tensión** Este aspecto, que también es parte del proceso de cálculo, está sujeto a restricciones técnicas dictadas por el profesional, así como a las directrices del Código Nacional de Electricidad–Utilización<sup>1</sup> en la *Sección 050–102*, donde se establecen los límites a considerar (Ministerio de Energía y Minas, 2006). La relación con el funcionamiento se basa en garantizar la correcta tensión eléctrica suministrada a los equipos en la instalación, asegurando así la vida útil adecuada y cumpliendo a su vez con el criterio de seguridad.

$$\Delta V_{max} \leq 4\% \quad (3.1)$$

Donde:

$\Delta V_{max}$  : Caída de tensión máxima permitida desde el alimentador hasta el último punto de utilización.

El límite expuesto en la Ecuación 3.1 es un valor máximo referencial para todo el conjunto de la instalación; no obstante, la norma también permite limitarlo a  $\Delta V_{max} \leq 2,5\%$  en alimentadores y circuitos derivados.

<sup>1</sup> Se puede consultar el documento *online* en: [http://www.minem.gob.pe/\\_detalle.php?idSector=6&idTitular=626&idMenu=sub114&idCateg=340](http://www.minem.gob.pe/_detalle.php?idSector=6&idTitular=626&idMenu=sub114&idCateg=340)

**3.2.2.3.3 Materiales & Equipo** En concordancia con el criterio de funcionamiento, la instalación debe contar con materiales y equipos que cumplan con las regulaciones vigentes. La normativa también establece los requisitos para el dimensionamiento y protección adecuada del equipo que forma parte de la instalación.

### **3.2.3 Flujo de Trabajo Convencional**

Si bien el concepto de flujo de trabajo es subjetivo y no tiene un carácter definitivo, cada profesional establece su propio método para abordar el diseño y cálculo de las instalaciones eléctricas. En la presente tesis, el autor establece un flujo de trabajo que califica como “convencional”, basado en su experiencia personal y la colaboración con otros profesionales de la especialidad.

#### **3.2.3.1 Etapa de Estudio**

En esta primera etapa, los profesionales suelen estudiar detenidamente la arquitectura de la edificación. El objetivo es comprender las necesidades generales del proyecto, anticipar posibles inconvenientes durante el diseño y considerar los alcances normativos pertinentes. También se tienen en cuenta las necesidades particulares del propietario.

#### **3.2.3.2 Etapa de Diseño**

La etapa de diseño implica el uso de herramientas informáticas que han reemplazado al método tradicional de “papel y lápiz”. El flujo de trabajo convencional se basa en el algoritmo que cada profesional considera más adecuado para distribuir cargas y puntos de alimentación eléctrica.

En este contexto, el Dibujo Asistido por Computadora (CAD) adquiere gran relevancia al permitir la rápida corrección y creación de múltiples propuestas de diseño. Estas propuestas pueden ser estudiadas en conjunto con otros profesionales de la especialidad o de áreas relacionadas.

El uso de archivos de dibujo (por ejemplo, con extensión *.dwg*) que sean intercambiables entre profesionales es crucial en esta etapa, ya que la arquitectura del proyecto determina el desarrollo del diseño de las instalaciones eléctricas. La gestión adecuada de archivos garantiza que se puedan manejar múltiples archivos de la misma especialidad bajo una nomenclatura preestablecida.

Aunque el software CAD actual ofrece diversas formas de gestionar archivos, se debe analizar la eficiencia de este enfoque en el flujo de trabajo convencional, ya que no siempre proporciona la máxima seguridad.

El flujo de trabajo común en esta etapa comprende los siguientes pasos:

1. Ubicar puntos de suministro, como acometida general y tableros de distribución.

2. Ubicar puntos de alumbrado, tomacorrientes, cargas de fuerza, pozo a tierra y pararrayos de ser necesario.
3. Plantear el entubado o canalizado de circuitos & montante.
4. Realizar una evaluación previa rápida de las distancias críticas donde podrían presentarse caídas de tensión mayores a las permisibles.

En cuanto a los tres primeros puntos mencionados anteriormente, la *Norma Técnica EM.010-Instalaciones Eléctricas Interiores*, parte del Reglamento Nacional de Edificaciones (Ministerio de Vivienda, Construcción y Saneamiento, 2006), indica que los planos o láminas generados deben estar bajo formatos normalizados. Aunque la norma no establece obligaciones sobre la representación gráfica bidimensional o tridimensional, se recomienda que ciertos aspectos de la instalación eléctrica tengan su propio *Plano de Detalle*.

En consecuencia, el algoritmo utilizado en el flujo de trabajo convencional implica que los planos de instalaciones eléctricas comúnmente se representan en dos dimensiones. Aunque esta forma de representación no tiene falencias en su interpretación, la obtención de una vista tridimensional (axonométrica) ofrecería una mejora significativa y una mayor comprensión de la visión propuesta por el ingeniero electricista.

### **3.2.3.3 Etapa de Cálculo**

En la etapa de cálculo, según el flujo de trabajo convencional, se procede a realizar los cálculos eléctricos pertinentes. En esta fase, se evalúa en detalle la carga instalada y se diseña el equipo de protección, así como los aspectos técnicos relacionados con los conductores y entubados. La forma en que se abordan estos cálculos puede variar según el ingeniero proyectista, pero siempre se tienen en cuenta los criterios expuestos en la sección 3.2.2.

En cuanto a la documentación generada en esta etapa, se incluyen los planos generados y las tablas de cálculo que justifican los aspectos técnicos, el equipo eléctrico seleccionado y los accesorios dimensionados. Además, se incorporan los conceptos de cantidades y presupuestos basados en el proyecto. Esta información es relevante para la metodología BIM, ya que entra en juego dentro del estudio de la dimensión correspondiente (Ver Tabla 2.1).

### **3.2.3.4 Limitaciones**

Dentro del flujo de trabajo expuesto anteriormente, no se niega la funcionalidad de este y de las herramientas que hacen posible su desarrollo, más en sí mismo es viable diagnosticar falencias que se pueden hallar en un estudio exhaustivo, tanto desde la gestión de bases arquitectónicas o gestión de interferencias entre especialidades, siendo esta última un aspecto donde se encuentran

inconvenientes a niveles de diseño e incluso agravándose cuando el proyecto se encuentra en etapa de ejecución.

Si bien el flujo de trabajo expuesto anteriormente y las herramientas utilizadas son funcionales y permiten llevar a cabo el diseño y cálculo de instalaciones eléctricas, es importante reconocer que también presentan algunas limitaciones. En primer lugar se hallan especialmente la gestión de bases arquitectónicas y la gestión de interferencias entre especialidades. Estas limitaciones pueden ocasionar problemas a nivel de diseño e incluso empeorar durante la etapa de ejecución del proyecto.

En la gestión de bases arquitectónicas, se pueden encontrar dificultades en la interpretación y manejo de los archivos de dibujo intercambiables entre profesionales. Aunque existen formatos normalizados y recomendaciones; la coordinación y compatibilidad entre diferentes software y versiones puede ser un desafío.

Por otro lado, la gestión de interferencias entre especialidades, como por ejemplo con la instalación eléctrica y otros sistemas (HVAC, fontanería, etc.), puede resultar complicada. Los errores en esta etapa pueden generar problemas durante la construcción y afectar la eficiencia del proyecto.

Abordar estas limitaciones y buscar soluciones que mejoren la coordinación y comunicación entre los profesionales y las diferentes especialidades involucradas en el proyecto se vuelve vital.

La implementación de metodologías BIM puede ser una alternativa para optimizar el flujo de trabajo y reducir las posibles interferencias y errores en la fase de diseño y construcción.

### **3.3 Características BIM**

Entendido el enfoque de especialidad que se requiere para el desarrollo de Diseño y Cálculo de Instalaciones Eléctricas Interiores, la metodología BIM y el análisis de sus respectivas características requieren un enfoque de la misma forma. En tal escenario, a continuación se expone un tratamiento aplicado a la ingeniería eléctrica para la metodología a aplicarse; ya que, a diferencia de otras especialidades, dentro del conjunto de conocimientos que se demandan para el BIM, este llega a configurarse de una forma mucho más técnica, por lo que el entender la interrelación de estas características se hace de vital importancia para una posterior adaptación a una herramienta informática en específica y por ende, las consideraciones que se han de tener para un eventual programado.

#### **3.3.1 Necesidad**

Dentro del proceso de inserción al BIM, una característica fundamental a considerar es la comprensión de las necesidades específicas para el desarrollo exitoso del diseño y cálculo de instalacio-

nes eléctricas de un proyecto. Según Oh et al. (2015, p. 198), esta etapa se conoce como el *Pre-diseño* en el contexto del BIM.

En el caso del sistema eléctrico de utilización, el estudio de las necesidades se enfoca en la fase de planificación y contempla los siguientes puntos:

- Tipo de edificación.
- Características arquitectónicas.
- Alcances y requisitos del propietario.

Estos tres aspectos están estrechamente relacionados y son cruciales para definir las demandas normativas y técnicas que deben cumplirse para garantizar el correcto funcionamiento del sistema eléctrico, en armonía con otras especialidades que forman parte del proyecto. El tipo de edificación define las necesidades específicas que deben ser satisfechas técnicamente, ya que diferentes edificios tienen requisitos eléctricos distintos según su uso y características.

Por otro lado, las características arquitectónicas son un aspecto más detallado que permite planificar desde la carga eléctrica a instalarse hasta la ubicación del equipo eléctrico en función de los espacios arquitectónicos. Es en esta etapa donde pueden surgir modificaciones técnicas para adaptar el diseño eléctrico a las particularidades de la edificación.

Finalmente, las necesidades y requerimientos del propietario, que pueden no estar completamente normados, pero son fundamentales para definir lo que es técnicamente posible y permitido dentro del marco normativo. Entender y delimitar estas necesidades contribuye a una planificación adecuada y una implementación exitosa del BIM en el proyecto de instalaciones eléctricas.

### **3.3.2 Información**

La característica de información es fundamental en BIM, ya que es precisamente en este aspecto donde se destacan las diferencias con el enfoque convencional de CAD (Dibujo Asistido por Computador).

En BIM, la información juega un papel esencial y se presenta de manera estructurada y detallada. El nivel de desarrollo de la información (LOD - Level of Development) varía según la cantidad y calidad de los datos proporcionados para cada objeto o componente del modelo. Por ejemplo, en el caso de un punto de luz, el LOD aumentaría a medida que se proporciona información detallada como:

- Posición geométrica en el proyecto.
- Denominación respecto al interruptor de control.

- Tensión de Operación.
- Potencia Nominal.
- Flujo Luminoso.
- Fabricante.
- Costo.
- Tipo de Montaje.
- Accesorios Adicionales.
- Etc.

El nivel de detalle y precisión de la información aumenta con cada dato adicional proporcionado, lo que se traduce en un mayor LOD BIM para el objeto (Figura 3.1). Es importante tener en cuenta que, a medida que se mejora el LOD, se obtiene una mayor comprensión y gestión de los componentes del proyecto eléctrico.

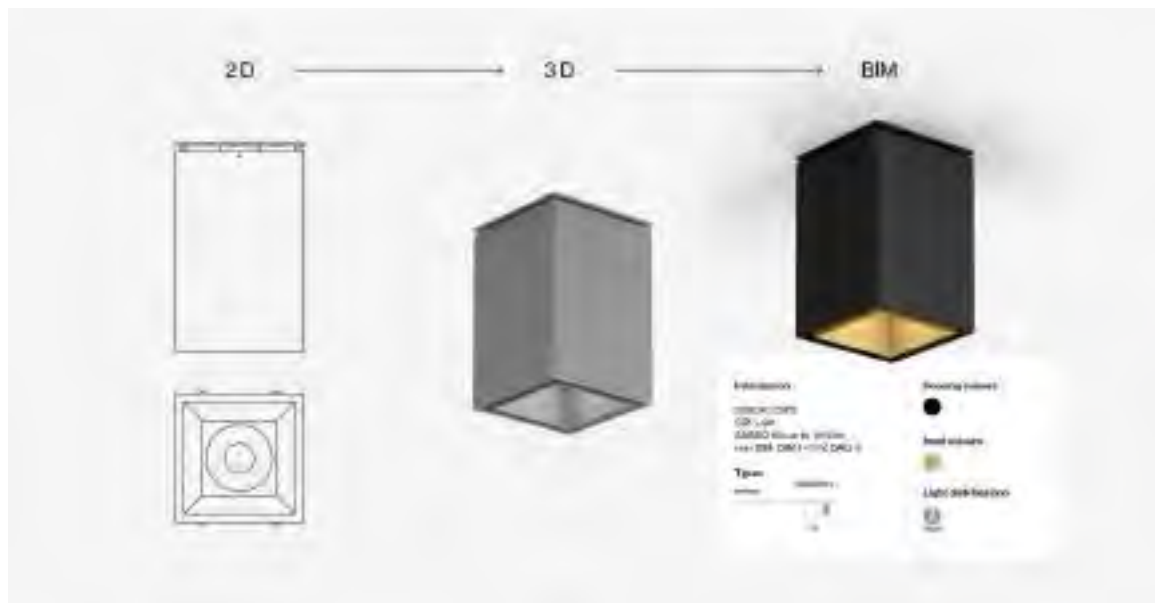
En la Figura 3.1 se muestra una comparativa entre el enfoque 2D, 3D y BIM. Mientras que en el enfoque 2D se tiene una representación bidimensional sin información detallada, en el enfoque 3D se incorpora la tercera dimensión, pero aún falta información relevante. Por otro lado, en BIM se logra una representación tridimensional completa y una amplia cantidad de información asociada a cada elemento del modelo, lo que permite una gestión más efectiva y una visualización realista del proyecto.

### **3.3.3 Gestión**

La característica de Gestión en BIM se refiere a la capacidad de los modelos para ser coordinados y gestionados de manera eficiente. Anteriormente, los modelos solían ser corregidos de forma independiente, lo que dificultaba la coordinación entre las diferentes disciplinas involucradas en el proyecto. Sin embargo, con BIM, se logra una integración más efectiva entre todas las partes del proyecto, lo que facilita la colaboración y la toma de decisiones.

La gestión en BIM se beneficia especialmente de las dimensiones adicionales que ofrece el enfoque 3D (ver Tabla 2.1). La capacidad de visualizar el proyecto en tres dimensiones permite identificar posibles conflictos y problemas de diseño antes de que ocurran en la construcción real. Esto permite realizar ajustes y correcciones de manera oportuna, lo que ahorra tiempo y recursos.

Además, la gestión en BIM también abarca aspectos relacionados con la sostenibilidad energética. La conciencia ambiental actual busca proyectos que sean sustentables y eficientes en el uso

**Figura 3.1***Comparativa 2D, 3D & BIM*

*Fuente:* Ilustración tomada de XAL Company (2022)

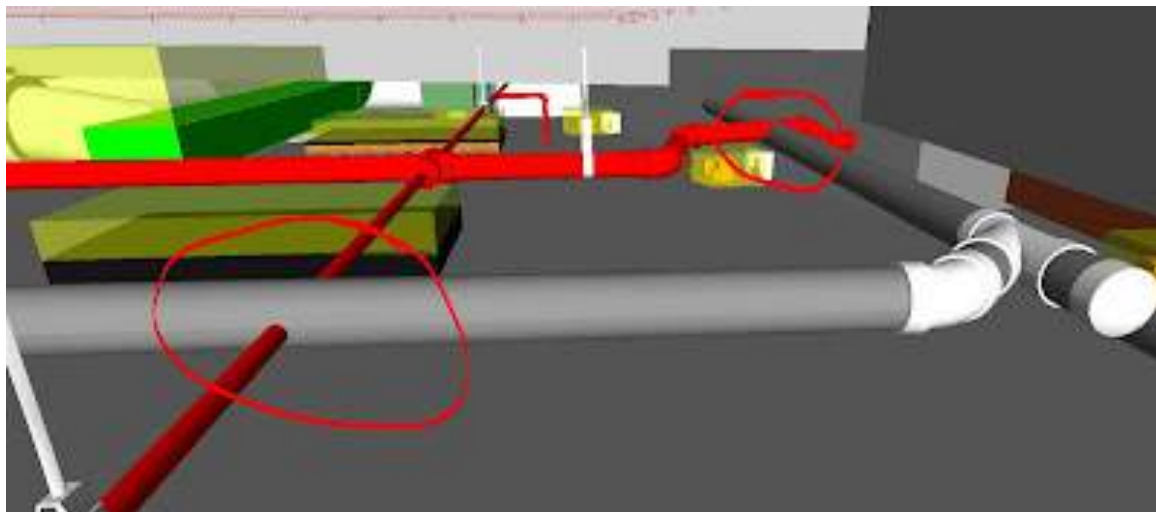
de la energía. En este sentido, el trabajo del ingeniero electricista adquiere una mayor relevancia en los roles BIM, ya que su experiencia es fundamental para asegurar que el proyecto cumpla con los estándares de sustentabilidad energética.

En cuanto a la complejidad de la característica de Gestión en BIM, es importante mencionar dos aspectos clave que Trebbi et al. (2019, pp. 115–116) destacan:

### **3.3.3.1 Detección de interferencias o *Clash Detection***

El término *Clash Detection*, traducido como Detección de Interferencias, es una característica esencial de la gestión BIM. En el contexto de un proyecto integral, es fundamental que todas las especialidades que lo componen puedan coexistir sin interferencias geométricas o técnicas entre ellas. Es común que durante el diseño de instalaciones eléctricas, se puedan producir casos en los que los conductos eléctricos se crucen con elementos estructurales o instalaciones sanitarias, como se muestra en la Figura 3.2.

La metodología BIM permite abordar este problema de manera proactiva, al priorizar la coordinación en lugar de la corrección. El *Clash Detection* es una herramienta que facilita la identificación temprana de posibles interferencias entre las distintas especialidades del proyecto. Esto se logra mediante el análisis de los modelos en 3D, donde se detectan áreas de conflicto y se generan informes que muestran las colisiones potenciales.

**Figura 3.2***Clash Detection en BIM*

Fuente: Imagen tomada de CAD Outsourcing-Engineering Design Services (2018)

### 3.3.3.2 Verificación normativa o *Code Checking*

El *Code Checking* o Verificación Normativa es otro aspecto relevante de la gestión BIM. Esta característica se enfoca en garantizar que el proyecto cumpla con todas las normas y regulaciones vigentes, tanto a nivel local como nacional, que son aplicables a la especialidad de instalaciones eléctricas.

El uso de la metodología BIM facilita la verificación normativa, ya que permite integrar la información normativa directamente en el modelo 3D. Los profesionales pueden acceder a bases de datos actualizadas de normativas y regulaciones eléctricas y aplicarlas automáticamente al modelo. Esto agiliza el proceso de comprobación de cumplimiento normativo y ayuda a evitar errores involuntarios que puedan surgir en el diseño.

## 3.4 Flujo de Trabajo BIM como Algoritmo

El desarrollo de tareas en el ámbito de metodologías tradicionales o modernas siempre requiere de un Workflow (Wf) o *Flujo de Trabajo*, que sirve como guía para el desarrollo de proyectos.

En el caso específico del diseño y cálculo de instalaciones eléctricas interiores con la metodología BIM, el flujo de trabajo se asemeja a la formulación de un algoritmo, ya que ambos implican una serie de pasos para ejecutar tareas.

Actualmente, en el sector privado, que lleva la delantera en la implementación y uso del BIM, cada empresa aplica flujos de trabajo BIM que han sido propuestos por ellas mismas, adquiridos de



terceros o tomados de autores especializados en el tema. Es importante destacar que el planteamiento de flujos de trabajo constituye propiedad intelectual, similar a las propuestas de algoritmos en el ámbito de la programación digital. Por lo tanto, para esta tesis se ha optado por utilizar un flujo de trabajo previamente elaborado con fines académicos, que permita alcanzar los objetivos propuestos.

Según la propuesta de Chen et al. (2016), el flujo de trabajo BIM para la aplicación de la especialidad de electricidad en el desarrollo del proyecto se divide en seis fases principales:

1. Preparación del diseño
2. Diseño eléctrico
3. Coordinación integral
4. Generación y ajuste de vistas 2D
5. Revisión
6. Entrega

El primer paso del flujo de trabajo propuesto por Chen et al. es la “Preparación del diseño”, que involucra etapas de estudio de necesidades para el diseño eléctrico. En este sentido, se observa cierta similitud con el flujo de trabajo tradicional mencionado previamente en el punto 3.2.3 de este capítulo. No obstante, Chen et al. destaca que su enfoque se basa en la metodología BIM, lo que implica un alejamiento de la metodología tradicional. Además, señala que las etapas restantes de su propuesta no necesariamente deben ser consecutivas, sino que todas funcionan de manera simultánea en el flujo de trabajo.

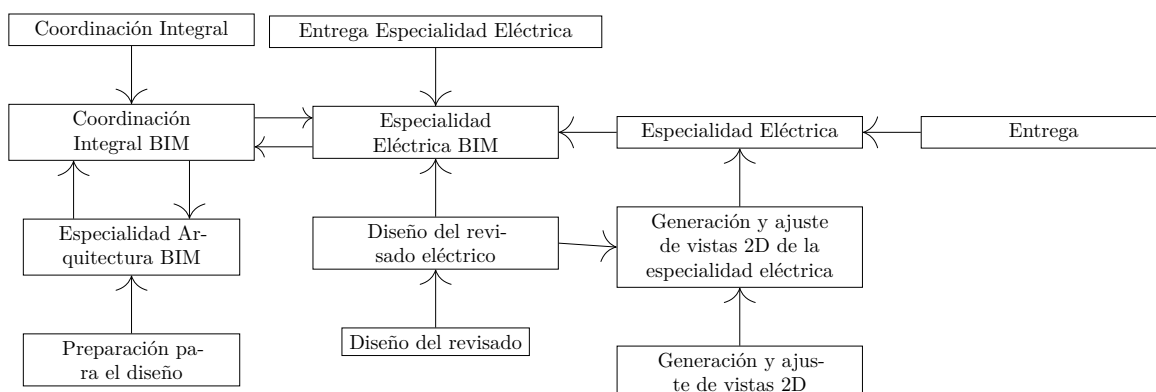
El flujo de trabajo BIM propuesto por Chen et al. para la especialidad eléctrica se muestra en la Figura 3.3. Este flujo de trabajo no sigue una naturaleza secuencial típica de un diagrama de flujo en algorítmica. En lugar de eso, Chen et al. explica que su enfoque integra tanto la operación de roles BIM como la aplicación adecuada del software. Primero se realiza la planificación con los roles y luego se procede a aplicarla en el software correspondiente.

Considerando las características BIM mencionadas en el numeral 3.3, es posible ampliar la propuesta de Chen et al. para que incluya estas características en su flujo de trabajo. Sin embargo, es importante destacar que estas características no forman parte del flujo de trabajo de manera secuencial, sino que están presentes en todas las etapas simultáneamente.

La Figura 3.4 muestra cómo las características BIM (necesidad, información y gestión) se integran en el flujo de trabajo propuesto por Chen et al. La necesidad (sombreado marrón) se refiere a los alcances de estudio y diseño que involucran tanto la especialidad eléctrica como la arquitectura del proyecto. La información (sombreado verde) no es mencionada explícitamente, pero está presente

**Figura 3.3**

*Workflow BIM de Chen et al. para la especialidad eléctrica (Adaptado)*

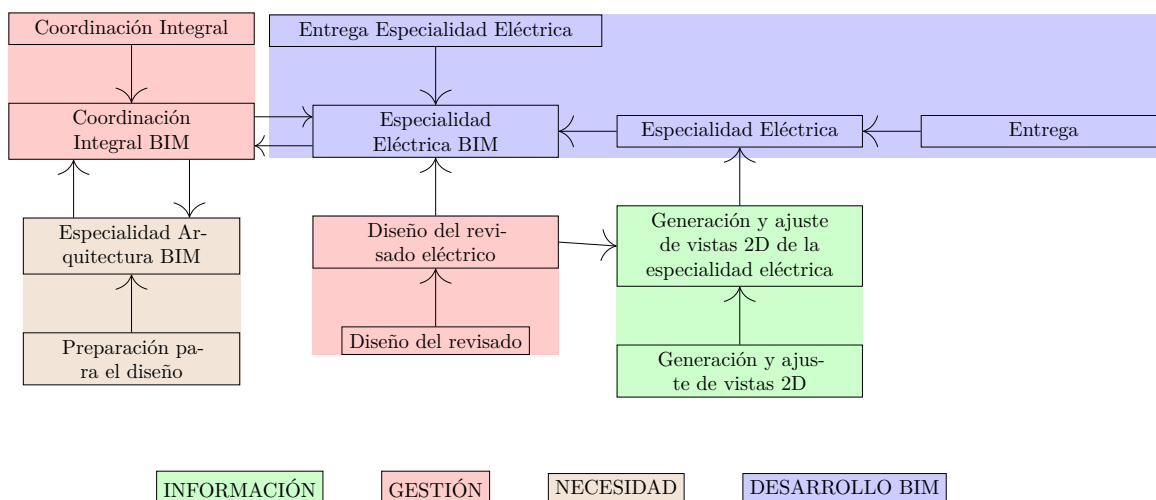


*Fuente:* Adaptado a partir de Chen et al. (2016, p. 761)

en la generación de modelos 3D y en la obtención de vistas 2D a partir de estos modelos. La gestión (sombreado rojo) es de gran importancia en el desarrollo BIM del proyecto. Se evidencia en la etapa de coordinación integral, que incluye la detección de interferencias (*Clash Detection*), y en la revisión, que se relaciona con el cumplimiento de las normativas (*Code Checking*).

**Figura 3.4**

*Características BIM en el Workflow BIM de Chen et al. (Adaptado)*



*Fuente:* Elaboración propia a partir de Chen et al. (2016, p. 761)

Finalmente, se debe comprender que el flujo de trabajo BIM en cualquiera de sus especialidades o como integración completa a un proyecto, no compone en sí mismo a un algoritmo “literal”, funciona de una forma mucho más amplia, debido a que a comparación de tradicionales métodos de trabajo; BIM explica su metodología a través de la total interacción entre los profesionales que son parte del proceso de diseño y cálculo de instalaciones eléctricas.

### 3.5 Necesidad, Información & Gestión en Software BIM

En este capítulo se ha evidenciado que el BIM requiere de herramientas informáticas para su aplicación, por lo que es importante relacionar las características expuestas anteriormente. En el segundo capítulo de la tesis se mencionaron algunos desarrolladores de software BIM que hacen posible esta aplicación. A continuación, se procederá a evaluar el software BIM más adecuado en base al análisis de las características previamente mencionadas para el desarrollo de instalaciones eléctricas interiores.

#### 3.5.1 Elección del Software BIM

La Figura 3.5 resume la diferencia entre las características BIM y su implementación en el software. Es importante destacar que el uso del software BIM requiere una ejecución manual para lograr un diseño completo con cálculos precisos, considerando la verificación del cumplimiento del *Code Checking*, como se explicó en la sección anterior. Es fundamental tener claro que BIM y Software BIM son conceptos distintos, donde BIM se refiere a la metodología y Software BIM es la interfaz de aplicación de dicha metodología.

**Figura 3.5**

*Necesidad, Información & Gestión en Software BIM*



*Fuente:* Elaboración propia

La Tabla 3.1 muestra una comparativa respecto a los principales exponentes BIM en relación al desarrollo de la especialidad de instalaciones eléctricas. Esta referencia destaca que no todos los desarrolladores de Software BIM incluyen en su funcionalidad la especialidad de instalaciones eléctricas. Esta evidencia subraya la importancia de elegir cuidadosamente el software en el que se desarrollará un proyecto, ya que, a diferencia de un flujo de trabajo tradicional con CAD, donde solo

se enfoca en el desarrollo vectorial de líneas, BIM requiere que el programa incluya características BIM específicas para el diseño y cálculo de instalaciones eléctricas interiores. Por lo tanto, la elección adecuada del software BIM asegurará la capacidad de abordar de manera efectiva y completa la especialidad de instalaciones eléctricas en un proyecto BIM.

**Tabla 3.1**

*Especialidad Eléctrica en Software BIM*

Software	Especialidad Eléctrica	Software	Especialidad Eléctrica
4MSA	SÍ	AECOSim Bentley	SÍ
Archibus	NO	ArchiCAD Graphisoft	SÍ
AutCAD Civil 3D	NO	Autodesk Fabrication	NO
Autodesk Navisworks	NO	Autodesk Revit	SÍ
Autodesk Robot	NO	Bentley Core	SÍ
BIM 360 Autodesk	NO	BIMCollab ZOOM	NO
Cype IFC Builder	NO	Cype CAD	NO
DesignBuilder	NO	EcoDomus	NO
FM:Systems	NO	Graitec Desing	NO
IBM Tririga	NO	IES Virtual	NO
Infracore	NO	Innovaya	NO
Keynetix	NO	AllPlan Nemetschek	Parcialmente
Scia Nemetschek	NO	Vectorworks Nemetschek	NO
Onuma System	NO	Primtech	SÍ
RhiNOBIM (Beta)	NO	Serraview	NO
Solibri Model Checker	NO	StructureSoft	NO
Synchro Bentley	NO	Tekla BIMSight	NO
Tekla Structures	NO	Vico Office	NO
ArchiFM for ArchiCAD	NO		

*Fuente:* Adaptado a partir de Gallardo y Guadalupe (2017)

La Tabla 3.2 proporciona una comparativa realizada por Sancha Rubio respecto a los principales software BIM y su aplicación en la especialidad de instalaciones eléctricas. Los indicadores analizados son las Bibliotecas, el Manejo, el Grado de Utilización y la Capacidad de Cálculo.

La importancia de los indicadores mostrados en la Tabla 3.2 nos relacionan a qué software se inclinaría normalmente un profesional dedicado al área técnica de ingeniería eléctrica en el sector constructivo.

Las Bibliotecas se refieren a la disponibilidad de elementos premodelados con información detallada. Algunos software BIM ofrecen una amplia variedad de elementos premodelados, lo que facilita el proceso de diseño y cálculo de instalaciones eléctricas.

**Tabla 3.2***Comparativa Software BIM*

	ArchiCAD Graphisoft	Autodesk Revit	Cype	Tekton 3D	
<b>Bibliotecas</b>	9	9	7	5	0-4 <i>Malo</i>
<b>Manejo</b>	7	8	2	6	4-6 <i>Regular</i>
<b>Grado de Utilización</b>	9	10	6	3	7-9 <i>Bueno</i>
<b>Capacidad de Cálculo</b>	NO	SÍ	SÍ	SÍ	10 <i>Muy Bueno</i>

*Fuente:* Adaptado a partir de Sancha Rubio (2022, p. 93)

El Manejo se relaciona con la facilidad y versatilidad para manipular circuitos, equipos eléctricos y conductos dentro del software. Un buen manejo facilita la gestión y coordinación de la especialidad eléctrica con las demás disciplinas del proyecto.

El Grado de Utilización indica la popularidad y preferencia de los profesionales en cuanto a la elección de software. En este caso, el producto de Autodesk, Revit, es el más utilizado y preferido entre los profesionales.

La Capacidad de Cálculo se refiere a la habilidad del software para realizar cálculos específicos en instalaciones eléctricas. Si bien tener esta capacidad es ventajoso, es importante tener en cuenta que los cálculos generados deben ser verificados y ajustados según la normativa del país donde se ejecutará el proyecto.

Con base en los indicadores mencionados, se destaca que Revit de Autodesk es la opción más recomendada para el desarrollo de proyectos de instalaciones eléctricas en BIM. Además, su compatibilidad con interfaces de programación y la disponibilidad de una amplia biblioteca digital de elementos para instalaciones eléctricas, hacen que sea una herramienta más completa y versátil en comparación con ArchiCAD de Graphisoft.

### 3.5.2 *Revit*

Revit, desarrollado por la empresa estadounidense Autodesk, es un software enfocado en el desarrollo BIM de modelos para la construcción. Aunque inicialmente se creía que BIM era exclusivamente para arquitectos y que Revit era simplemente una herramienta de dibujo automatizado, la comprensión de la metodología BIM ha demostrado que es mucho más que eso, abarcando diversas áreas de la ingeniería y la construcción.

La primera versión de Revit se lanzó en el año 2000, centrada en el diseño arquitectónico, pero con el tiempo se amplió su alcance para incluir ingeniería estructural y de instalaciones. Actualmente, Revit se ha subdividido en tres subprogramas: *Architecture*, para el diseño arquitectónico; *Structure*,

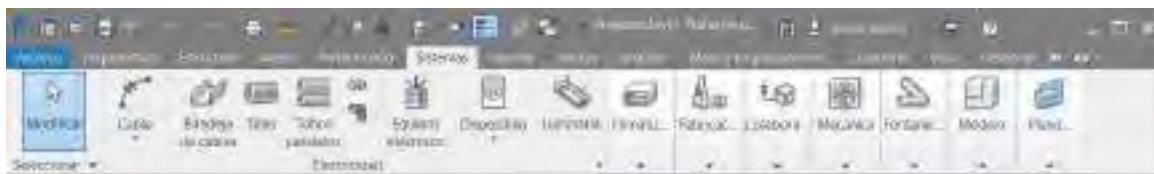
**Figura 3.6***Autodesk Revit Software*

*Fuente:* Icono de software tomado de Autodesk Latinoamérica (2022c).

para el modelado de elementos estructurales; y *MEP* (Mechanical, Electrical and Plumbing), que se enfoca en los sistemas de instalaciones mecánicas, eléctricas y de plomería.

En el caso específico de diseño de instalaciones eléctricas, se trabaja con el entorno *Electrical* dentro de la ficha de Sistemas de Revit. La herramienta ha evolucionado para integrar todas las especialidades en una sola instalación, lo que permite un desarrollo completo de la característica de Gestión BIM, y se puede apreciar en la interfaz de Revit, como se muestra en la Figura 3.7.

Es importante destacar que Revit ha logrado unificar la arquitectura, las instalaciones y la estructura como elementos únicos en el desarrollo de proyectos, lo que facilita la coordinación y la gestión en el proceso de diseño y construcción. Para el propósito de esta tesis, se enfocará únicamente en el entorno *Electrical* de Revit para el desarrollo de las instalaciones eléctricas interiores.

**Figura 3.7***Interfaz de Revit como producto integral*

*Fuente:* Captura de software propia.

### 3.5.2.1 El Entorno “Electrical”

El entorno *Electrical* de Revit, en cualquiera de sus versiones, es una ficha de especialidad diseñada para el desarrollo de instalaciones eléctricas en proyectos BIM. Aunque este entorno permite la interoperabilidad con otros software BIM, en muchos casos los profesionales optan por utilizar un

mismo software (o al menos del mismo desarrollador) para todo el proyecto, debido a las limitaciones de la interoperabilidad (ver Figura 3.8).

**Figura 3.8**

*El entorno Electrical de Revit*



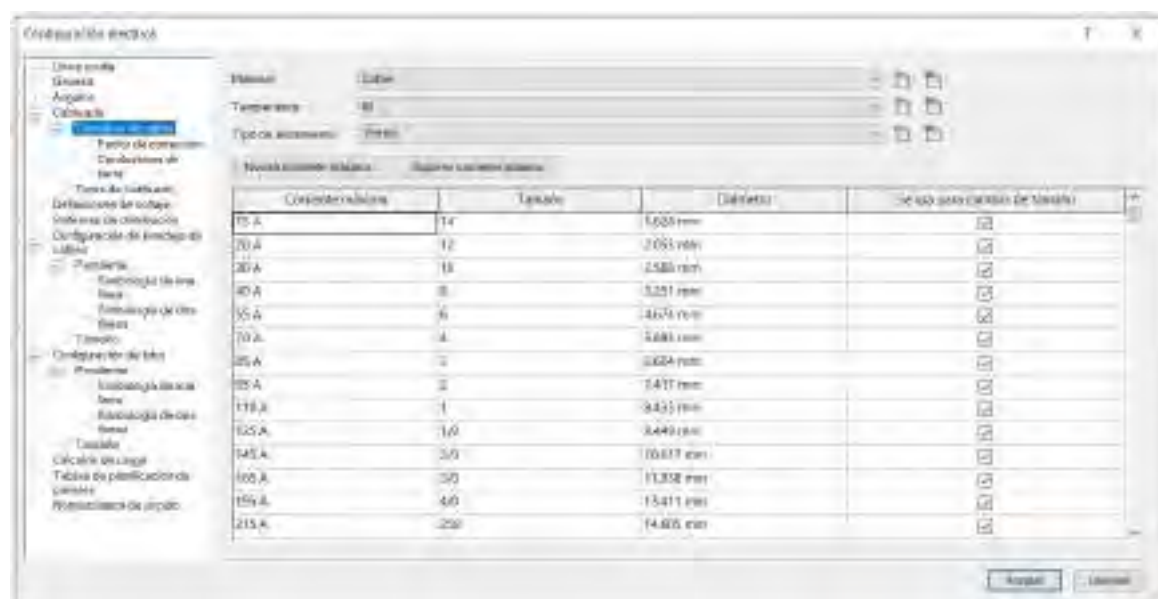
Fuente: Captura de software propia.

La característica principal que ofrece el entorno *Electrical* es la diferenciación clara entre el circuito cableado, los conductos y el equipo eléctrico a instalarse, lo que facilita un modelado básico del proyecto. Sin embargo, es importante destacar que estas opciones son solo las más generales para el diseño de instalaciones eléctricas, y para el cálculo técnico se requerirá explorar otras herramientas del software.

Adicionalmente, el entorno *Electrical* proporciona una paleta de configuración que permite realizar cálculos básicos de cargas, propiedades eléctricas de los conductores y aspectos físicos de los tubos o conductos a utilizar en el proyecto (ver Figura 3.9).

**Figura 3.9**

*Configuración Adicional del Entorno Electrical.*

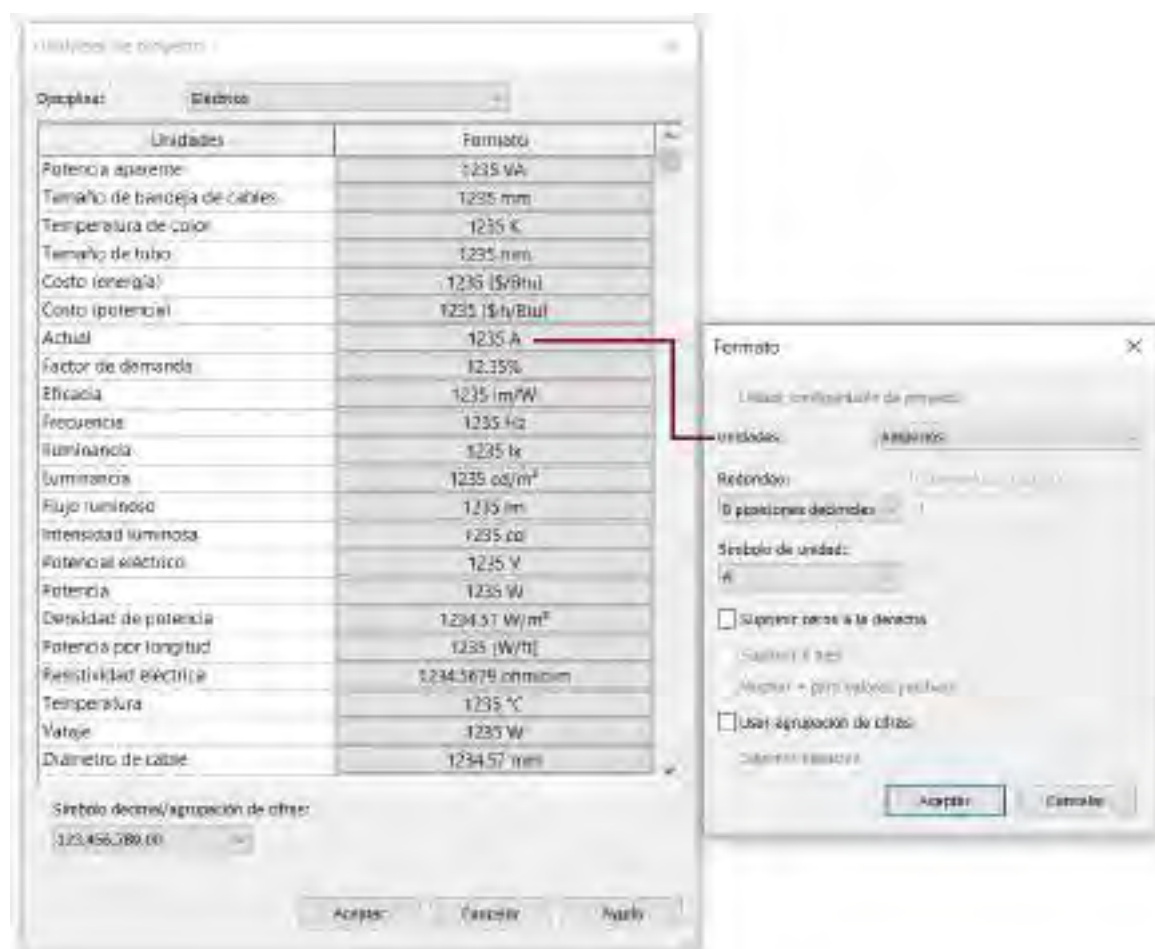


Fuente: Captura de software propia.

Es crucial manejar correctamente las unidades de medida en el software, y Revit permite realizar ajustes de unidades para adaptarse a las normas locales (Figura 3.10). Sin embargo, es necesario realizar una configuración adicional para cumplir con las unidades establecidas en la norma peruana («Ley 23560 – Sistema Legal de Unidades de Medida del Perú», 1982).

**Figura 3.10**

*Manejo de Unidades para disciplina Electrica en Revit.*



*Fuente:* Captura de software propia.

En la misma interfaz, también se puede observar otro inconveniente relacionado con las traducciones literales, como se muestra en la Tabla 3.3. Algunos términos en español pueden traducirse de forma errónea al inglés, lo que podría generar confusiones en la interpretación del usuario. Aunque estos errores de traducción no afectan la funcionalidad del software, es importante tenerlos en cuenta para evitar malentendidos y mejorar la precisión informativa en el manejo del programa.

Con el caso ejemplar visto en la Tabla 3.3, en todo el software se pueden hallar otros casos similares, y aunque esto no implica errores en la funcionalidad; dentro del manejo informativo del software podría configurar errores de interpretación por parte del usuario.



Tabla 3.3

*Errónea Traducción Literal. Caso: Corriente Eléctrica*

Español	Inglés	Traducción Literal
Corriente	<i>Current</i>	Actual

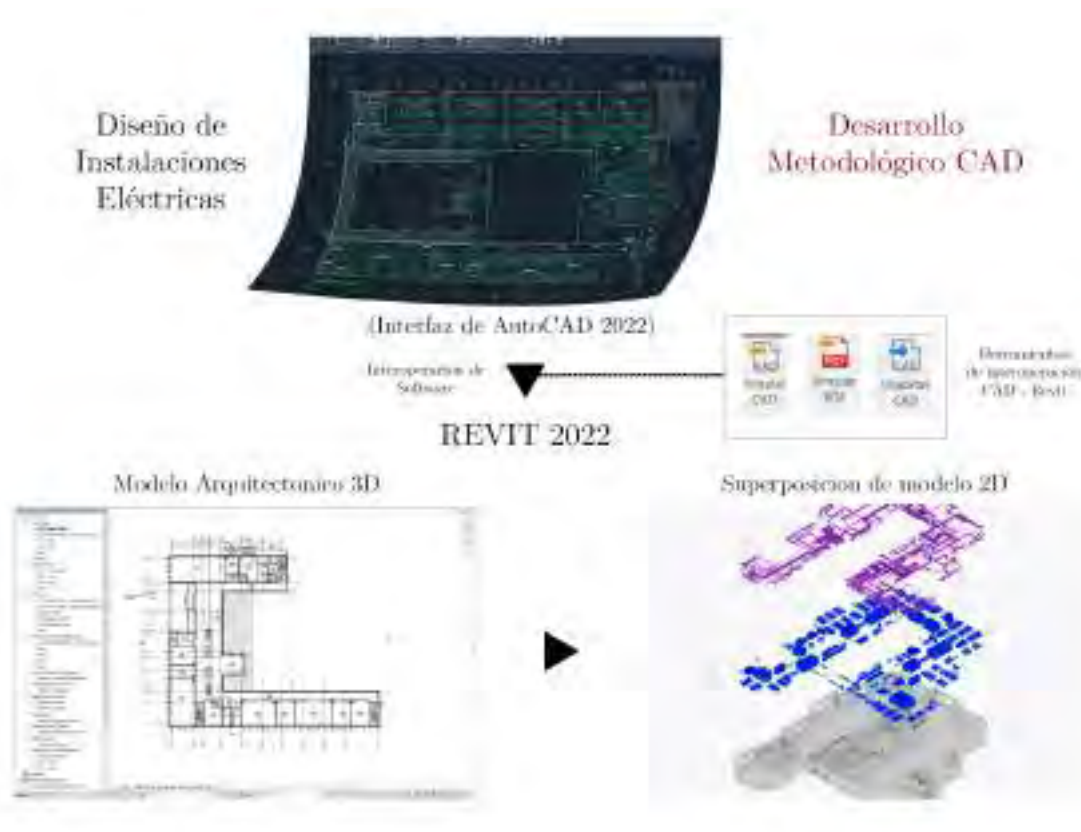
*Fuente:* Elaboración propia.

### 3.5.2.2 Uso convencional

Para entender el uso convencional que se le ha dado a Revit para el desarrollo de instalaciones eléctricas, se puede observar en la Figura 3.11 cómo se superponen dos flujos de trabajo. Esto sucede debido a que muchos profesionales todavía utilizan herramientas de dibujo asistido por computadora como AutoCAD para realizar el diseño preliminar de las instalaciones eléctricas. Luego, este diseño se importa o traspassa a Revit para continuar con el modelado en 3D y la generación de información BIM.

Figura 3.11

*Uso “Doble” Convencional Ilustrado*



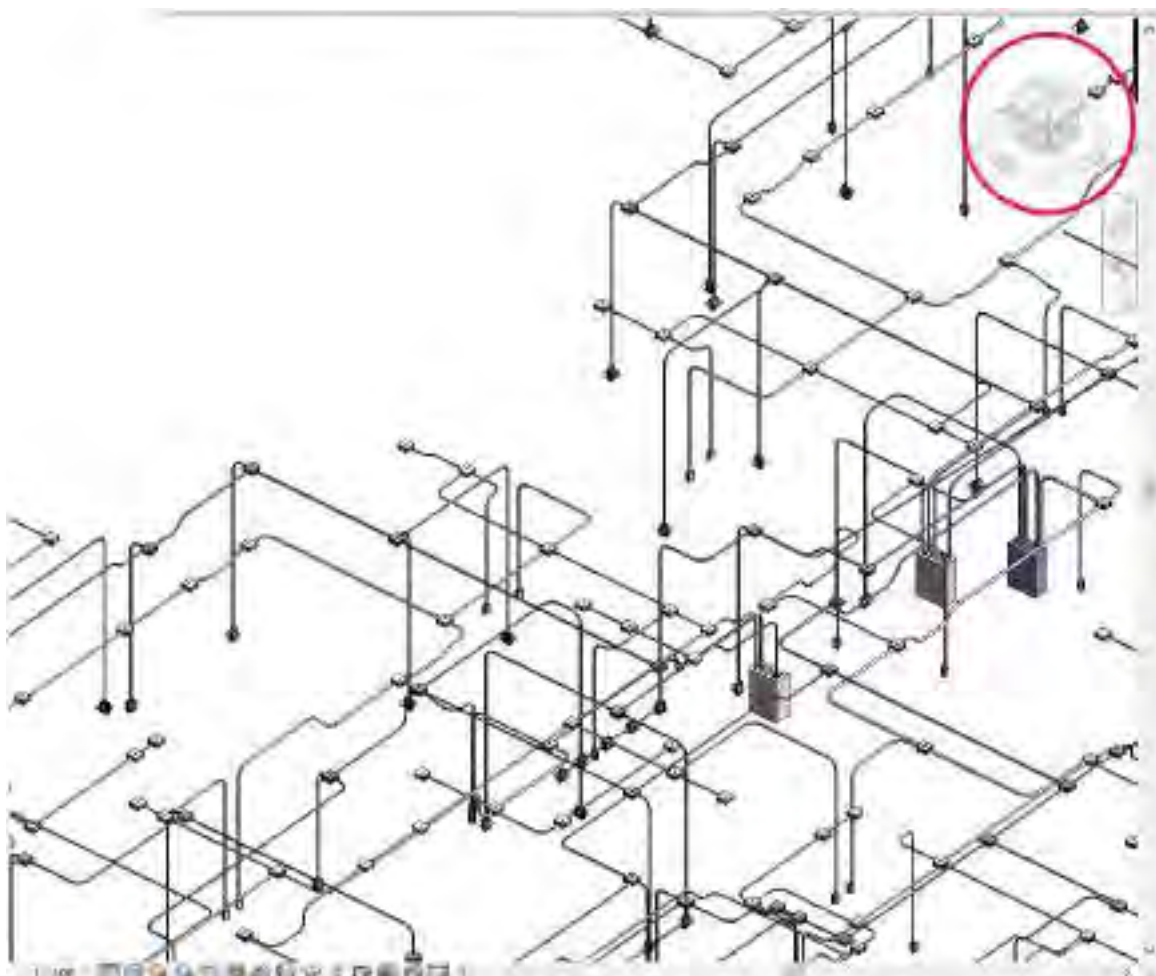
*Fuente:* Elaboración propia.

El uso convencional de Revit para instalaciones eléctricas presenta ciertas dificultades, especialmente en el modelado tridimensional de componentes como el equipo eléctrico y la ubicación de conductos. La interfaz de manejo en 3D, como el *ViewCube* (Ver Figura 3.12), puede resultar incómoda y poco intuitiva, lo que dificulta el modelado tridimensional eficiente. En lugar de utilizar estas herramientas, muchos profesionales optan por trabajar con vistas de elevación o cortes (Ver Figura 3.13), lo que puede ser más cómodo pero no aprovecha completamente el potencial del modelado 3D en Revit.

La superposición de flujos de trabajo y el uso convencional de Revit para instalaciones eléctricas pueden afectar la eficiencia y precisión del proyecto, ya que no se está aprovechando al máximo la metodología BIM. Es importante que los profesionales se familiaricen y se capaciten en el uso adecuado de las herramientas y funciones que ofrece Revit para instalaciones eléctricas, lo que permitirá un modelado más preciso y una mejor integración de la información en el proyecto.

### Figura 3.12

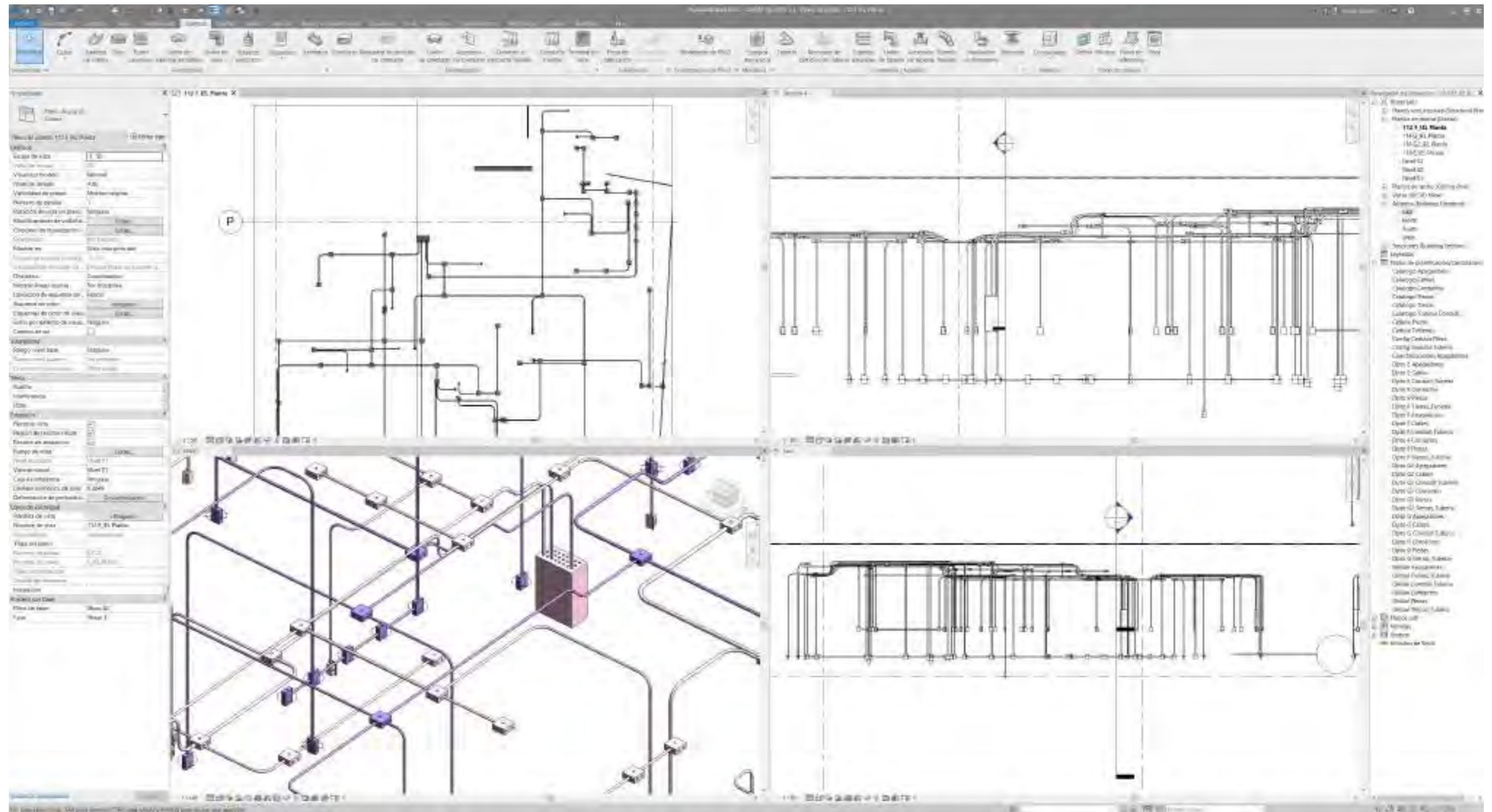
*Representación tridimensional de Modelo Complejo en Revit*



*Fuente:* Captura de software propia.

Figura 3.13

Vistas Auxiliares para el Modelado de Instalaciones Eléctricas en Revit



Fuente: Captura de software propia.

### 3.5.2.3 Modelado y cálculo de Instalaciones Eléctricas

Visto en la sección anterior se comprende que el entorno *Electrical* de Revit ofrece herramientas y funciones para el modelado y cálculo de instalaciones eléctricas dentro del contexto BIM. La eficacia y precisión en su uso dependen de la capacitación y experiencia del usuario, así como de la correcta interpretación y aplicación de las características BIM en el software. Es fundamental que los profesionales se familiaricen con las funcionalidades específicas de Revit para instalaciones eléctricas y adopten flujos de trabajo eficientes que maximicen los beneficios de la metodología BIM en este campo.

**3.5.2.3.1 Necesidad–Estudio** La característica de la necesidad en Revit se canaliza a través de la interfaz del programa. Existen varias formas de estudiar esta necesidad, siendo las más comunes: el premodelado arquitectónico directamente en el mismo software (ya sea en el mismo archivo o en diferentes archivos), o la interoperabilidad del BIM, que implica trabajar con información proveniente de otros programas o archivos, aunque esta opción puede conllevar el riesgo de perder detalles de información.

La Tabla 3.4 muestra diferentes opciones para abordar el estudio de necesidades eléctricas en Revit, considerando la capacidad de tener características premodeladas en el software.

**Tabla 3.4**

*Capacidad de tener Características Premodeladas en Revit*

Tipo de Vínculo	Formato de Archivo	Observaciones
<b>CAD</b>	<i>.dwg</i>	Requiere Modelado 3D adicional.
<b>BIM</b>		
Revit unico proyecto	<i>No aplica</i>	Archivo muy pesado
Revit diferentes archivos	<i>.rvt</i>	Requiere nomenclaturas de un Flujo de Trabajo
Otro Software	<i>.ifc</i>	Posible pérdida de información entre softwares.

*Fuente:* Elaboración propia.

Independientemente del enfoque seleccionado, en las versiones de Revit que contienen todas las especialidades en un solo programa, se puede desarrollar la característica BIM de forma visual y paramétrica.

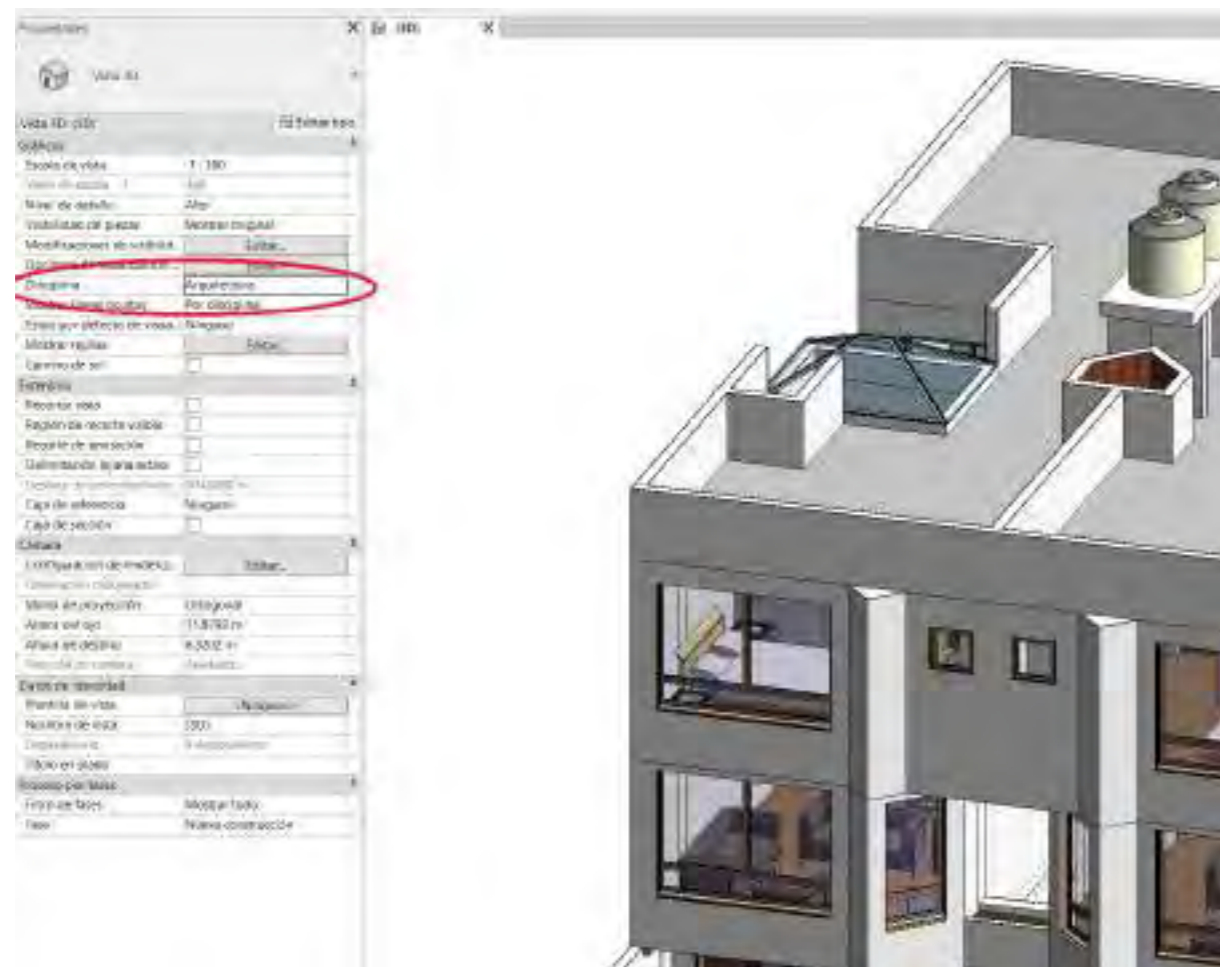
Las vistas del modelo pueden cambiarse para permitir el estudio desde diferentes perspectivas. Por ejemplo, la Figura 3.14a muestra una vista arquitectónica, donde se destacan elementos como

muros, puertas, ventanas y mobiliario, permitiendo analizar el diseño con todas las consideraciones del arquitecto proyectista. En caso de requerir información sobre elementos estructurales, como la ubicación precisa de estribos de acero, también es posible cambiar a la especialidad de Estructuras.

La Figura 3.14b, por otro lado, presenta una vista eléctrica, donde los elementos arquitectónicos se vuelven transparentes para enfocarse en los elementos sanitarios previamente modelados para el proyecto. Esto es especialmente relevante para analizar el emplazamiento del sistema de tuberías de agua y desagüe, ya que el estudio de necesidades eléctricas también implica considerar aspectos técnicos de otras especialidades.

Figura 3.14

Vistas de Revit para especialidades



(a) Vista Arquitectónica para Revit



(b) Vista Eléctrica para Revit

Fuente: Capturas de software propias.

Cabe mencionar que no todos los proyectos contarán con características premodeladas, por lo que será necesario realizar el modelado eléctrico desde cero en ciertos casos. Sin embargo, Revit ofrece herramientas y funcionalidades que facilitan este proceso y maximizan los beneficios de la metodología BIM en el campo de las instalaciones eléctricas.

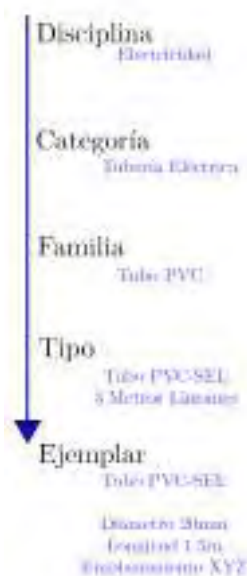
Esta funcionalidad discreta de cambiar entre especialidades, como se muestra en las vistas de la Figura 3.14, demuestra cómo un parámetro manipulable dentro del software es la disciplina o especialidad que se va a desarrollar, incluso en vistas macro del modelo BIM completo.

**3.5.2.3.2 Información–Familias Revit** En Revit, la característica informativa del BIM se aborda a través de su jerarquía de elementos, y es importante comprender esta estructura para aprovechar al máximo la capacidad del software. Aunque comúnmente se ha difundido erróneamente que los elementos en Revit simplemente se denominan "Familias", en realidad, es necesario entender la correcta denominación y jerarquía de los elementos para comprender sus capacidades informativas y de manipulación de parámetros.

La jerarquía de información dentro de Revit se muestra en la Figura 3.15. Esta estructura permite diferenciar a los elementos de forma mucho más detallada que en el CAD, donde simplemente se tiene un bloque vectorial de dibujo que puede requerir interpretación adicional. En Revit, un elemento puede contener una amplia cantidad de información, desde la disciplina a la que pertenece hasta detalles físicos y su ubicación en el proyecto.

**Figura 3.15**

*Jerarquía Informativa dentro de Revit*



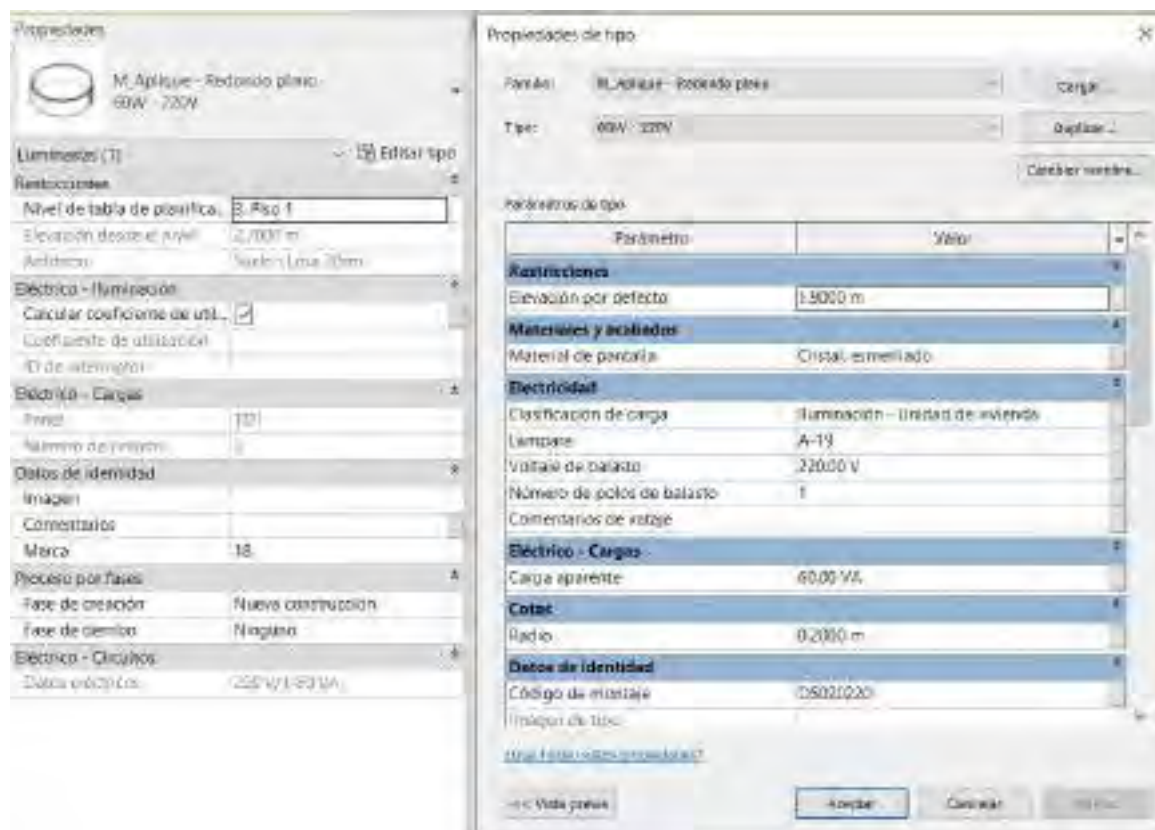
*Fuente:* Elaboración propia.

Para ilustrar este enfoque, consideremos el ejemplo de un tubo de PVC en una instalación eléctrica. En el CAD, bastaría con una línea que represente el tubo en un plano, y la información adicional requeriría anotaciones o documentos de especificaciones técnicas. En cambio, en Revit, el mismo tubo PVC puede contener información detallada, incluyendo su clasificación eléctrica, aspecto físico y ubicación específica en el proyecto.

La capacidad de manejo de elementos en Revit permite diferenciarlos a través de la clasificación de “Tipo” y “Ejemplar”. Esto posibilita la parametrización de elementos de acuerdo con sus características específicas. Por ejemplo, en la Figura 3.16, se muestra la ficha de propiedades de un artefacto de iluminación en Revit. En la ventana de "Propiedades de tipo", se pueden manipular una amplia gama de propiedades que son vitales para el cálculo de un proyecto de instalaciones eléctricas, como elevación, clasificación de carga, voltaje y carga aparente.

**Figura 3.16**

*Ficha de Propiedades de un elemento de Revit*



*Fuente:* Captura de software propia.

Sin embargo, es importante tener en cuenta que las bibliotecas de Revit suelen manejar el dato de potencia eléctrica como “Potencia Aparente” y, por defecto, utiliza los “Volt-Amperios” como unidad de potencia. En el territorio peruano, según el SLUMP, la unidad de medida asignada para medir la potencia eléctrica es el Watt (W). Por lo tanto, se requiere ajustar las unidades en el entorno



*Electrical* (Figura 3.10) o implementar un parámetro explícito para “Potencia” que se ajuste a las definiciones del ingeniero electricista.

**3.5.2.3.3 Gestión–Extensión Revit** Revit, como software enfocado al BIM, posee un potencial de gestión para proyectos, aunque presenta ciertas limitaciones en comparación con las subcaracterísticas más resaltantes de la gestión. En su interfaz, Revit cuenta con una ficha llamada "Gestión"(ver Figura 3.17a), pero la interpretación del software respecto a esta característica va más allá y abarca otras funcionalidades, como la interacción de parámetros del proyecto, el uso de materiales, la ubicación geográfica del proyecto, el diseño generativo en arquitectura y la interfaz de programación llamada “Dynamo”<sup>2</sup>.

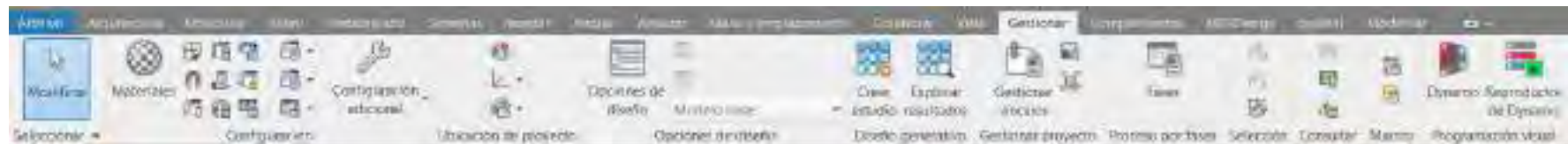
Para abordar las capacidades de gestión de *Clash Detection* y *Code Checking* en Revit, es necesario explorar otras fichas de la interfaz o recurrir a otro software complementario, como Navisworks. Dentro de Revit, las opciones que se ajustan a estos alcances se encuentran en las fichas “Analizar” y “Colaborar” (ver Figuras 3.17b y 3.17c). En las últimas versiones de Revit, que incluyen todas las especialidades en un mismo programa, estas fichas también contienen diversas opciones para diferentes disciplinas.

---

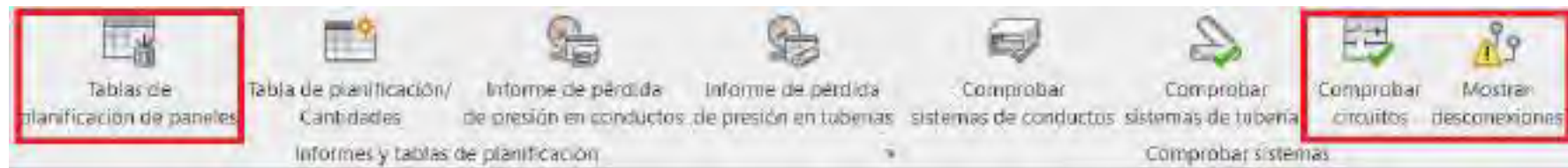
<sup>2</sup> Este último presta importancia para la presente tesis, no obstante, es en el siguiente capítulo en el cual se ahondará en esta herramienta.

Figura 3.17

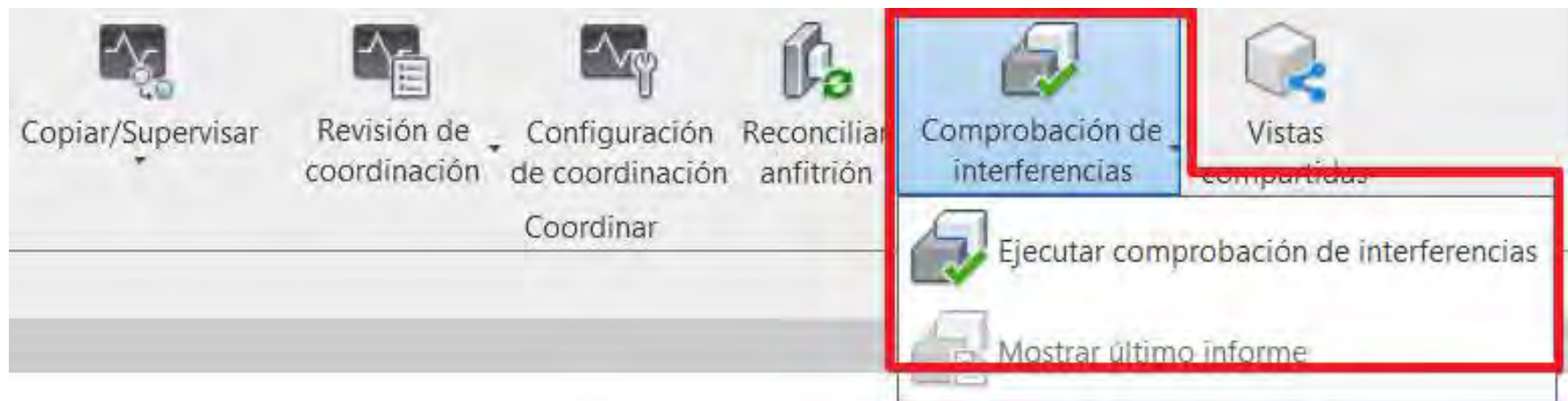
Funciones de Gestión dentro de Revit



(a) Gestión dentro de la interfaz de Revit



(b) Capacidad de análisis en instalaciones eléctricas en Revit.



(c) Posibilidad de Clash Detection en Revit

Fuente: Capturas de software propias.

Dentro del *Clash Detection*, la herramienta correspondiente presentada como “Comprobación de interferencias” funciona de una forma generalizada a todo el modelo, aunque cumple con lo que propone la característica; el reporte y el manejo de estas interferencias detectadas permite solamente un modo visual de gestión en esta forma.

Por otro lado, dentro del *Code Checking* no hay propuesta del software a permitir un manejo específico de la norma peruana. Revit también ofrece la posibilidad a comprobar circuitos, no obstante esta herramienta no es precisamente para comprobar caídas de tensión o sus límites (Ecuación 3.1); esto es cambio para comprobar que las conexiones eléctricas estén correctas a su punto de conexión (Ver Figura 3.18), el sistema (monofásico o trifásico) y otras compatibilidades que prevengan cortocircuitos.

### Figura 3.18

*Punto de Conexión Eléctrica en Revit*

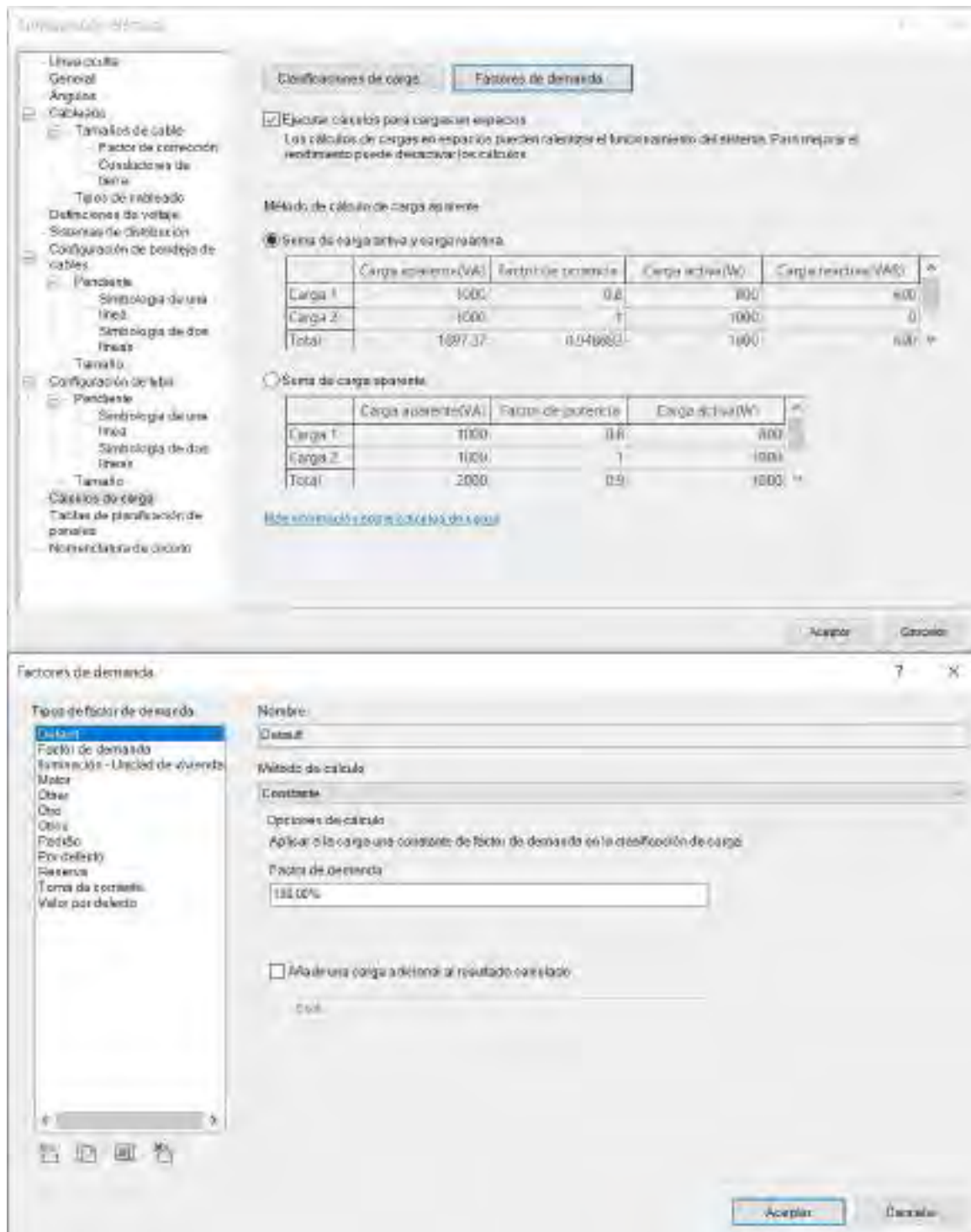


*Fuente:* Captura de software propia.

De la misma manera, junto a la revisión normativa en las cuales se sustente el cálculo de la carga, solamente se tiene disponible el uso de tablas de planificación, no obstante el cálculo que hace el software, podría calificarse de muy básico y sin contemplar del todo los alcances del CNE–Utilización dentro de la *Sección 050* para el planteamiento de cuadros de carga. La Figura 3.19 refiere de nuevo al entorno *Electrical*, ya que este cálculo funciona a base de configurar previamente todos los factores de demanda a utilizarse y del mismo modo, los métodos de cálculo de carga son solamente dos.

Figura 3.19

Interfaz para el cálculo de cargas en Revit.



Fuente: Captura de software propia.

La necesidad de una mejor implementación o ampliación de las herramientas de Revit se hace evidente para que se pueda interactuar con la normativa sin tener que explorar numerosas fichas y ventanas de configuración, lo que complica el uso del software BIM.

En caso de optar por la interoperabilidad de Revit con otro software BIM que abarque la característica de gestión, es posible realizar una migración. Sin embargo, es recomendable que ambos programas sean del mismo desarrollador para garantizar una mejor compatibilidad y flujo de trabajo. Navisworks, otro software de Autodesk, se enfoca en la gestión de proyectos BIM, incluyendo *Clash Detection* y dimensiones BIM de la planificación de tiempos.

Navisworks es un software enfocado a la metodología BIM vista desde el ámbito de la Gestión del proyecto del *Clash Detection* hasta dimensiones BIM de la planificación de tiempos.

### Figura 3.20

*Autodesk Navisworks Manage Software*



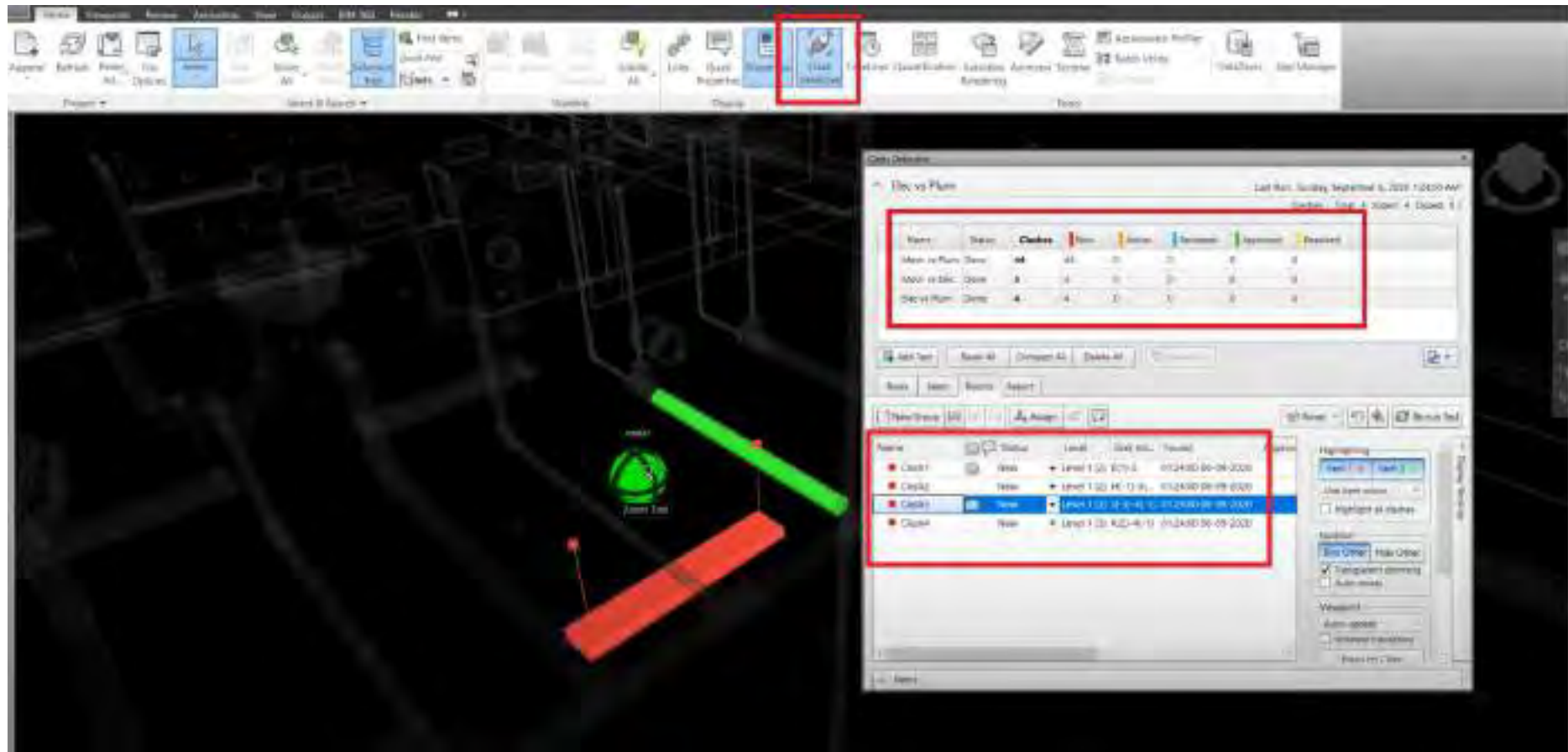
*Fuente:* Icono de software tomado de Autodesk Latinoamérica (2022b).

La herramienta de *Clash Detection* en Navisworks permite aprobar, revisar o subsanar todos los informes de interferencias detectadas. Además, al no ser un software dedicado al modelado y cálculo BIM, maneja archivos mucho más ligeros en peso que Revit (*.rvt*). No obstante, Navisworks no proporciona *Code Checking* asistido por la computadora, y el usuario debe realizar estas comprobaciones de forma manual.

La interfaz que se aprecia en la Figura 3.21 se tienen tres elementos resaltados. Primeramente la implementación dedicada de *Clash Detection* que posee Navisworks, seguidamente, dentro de la ventana dedicada, se presenta la capacidad de hacer las pruebas de interferencias entre disciplinas en específico; finalmente el reporte enseña el lugar preciso de las interferencias detectadas y si estas se subsanan o se aprueban, ya que, aún al año 2022, Navisworks no presenta la perfección en estas pruebas de interferencias.

Figura 3.21

Ventana de Clash Detection de Navisworks



Fuente: Captura de software propia.

#### **3.5.2.4 El tiempo de trabajo y su necesidad de mejora**

En el desarrollo anteriormente presentado, se ha puesto de manifiesto que el proceso de modelado manual de instalaciones eléctricas en el software BIM, Revit, representa un trabajo tedioso y que requiere una significativa inversión de tiempo y recursos. Cada elemento modelado exige un tratamiento especial debido a la variedad de parámetros necesarios para una implementación efectiva de la metodología BIM, a lo que se suma la complejidad adicional del manejo de múltiples vistas en Revit.

El proceso de modelado inicial demanda una atención minuciosa y repetitiva, lo que puede llevar varias horas, días o incluso semanas de trabajo para los profesionales involucrados. Aun para los más experimentados, es inevitable dedicar largas horas a tareas rutinarias y de alto esfuerzo cuando se realiza el modelado manual en Revit. Específicamente en el caso de las instalaciones eléctricas, los diseñadores se enfrentan a una serie de operaciones manuales necesarias, generando una carga de trabajo que podría ser considerablemente reducida.

Conscientes de la necesidad de disminuir el tiempo invertido en estas tareas, en el desarrollo posterior de esta tesis se evaluarán opciones para optimizar los tiempos de trabajo mediante una metodología innovadora en la especialidad de instalaciones eléctricas. El objetivo es transformar un proceso que actualmente consume horas de trabajo en una ejecución casi instantánea, representando un cambio significativo y prometedor en contraste con el modelado y cálculo manual en Revit.

## CAPÍTULO IV

### EVALUACIÓN DE LOS POTENCIALES DE LA PROGRAMACIÓN PARAMÉTRICA-VISUAL EN BIM

#### 4.1 Introducción

Dentro del campo de la ingeniería, especialmente en las especialidades eléctricas, electrónicas e informáticas, la formación profesional ha incorporado cada vez más conocimientos en programación. Sin embargo, en el ámbito del diseño y cálculo de instalaciones eléctricas, aún se tiende a realizar tareas de manera manual, ignorando el potencial que la programación podría ofrecer para optimizar y agilizar este proceso.

Aunque existen diversos software dedicados al diseño y cálculo de instalaciones eléctricas, desarrollar una nueva interfaz desde cero sería un trabajo complejo y demandante. Por tanto, el enfoque de la programación en este campo debe centrarse en mejorar las herramientas existentes, adaptándolas a los criterios y necesidades específicas de los ingenieros electricistas, en concordancia con las normativas nacionales.

En este capítulo, se evaluarán las posibilidades y el potencial de la programación para mejorar el diseño y cálculo de instalaciones eléctricas en el contexto del Software BIM. Se buscará comprender las limitaciones y oportunidades de las plataformas BIM disponibles para seleccionar la más adecuada a las necesidades del proyecto.

#### 4.2 Relación con el Software BIM

La programación paramétrica-visual se relaciona con el software BIM de manera similar a la programación tradicional. Sin embargo, la programación paramétrica-visual ofrece ventajas en términos de interacción y manipulación de elementos dentro del software BIM.

A diferencia de la programación tradicional, que puede aplicarse a cualquier software pero requiere un alto conocimiento de sintaxis y lenguajes de programación específicos, la programación paramétrica-visual es más accesible y permite una interacción más directa con los elementos del software BIM.



Es importante tener en cuenta que no todos los software BIM permiten la manipulación y programación abierta de sus herramientas. Algunos software solo permiten la interacción con ciertos lenguajes de programación, mientras que otros no permiten ninguna forma de manipulación.

En este contexto, los desarrolladores de complementos para software BIM buscan establecer vínculos directos que aseguren la compatibilidad o crean aplicaciones específicas para manipular el software, incluso cuando este no es completamente abierto a la programación.

En este sentido, la interfaz dedicada Dynamo se destaca como una opción prometedora para evaluar su integración con el software BIM – Revit. Dynamo ofrece una plataforma visual y paramétrica que facilita la manipulación y optimización de tareas en el diseño y cálculo de instalaciones eléctricas, lo que la convierte en una herramienta a considerar para mejorar la eficiencia y flexibilidad del software BIM.

### 4.3 Dynamo y su uso convencional

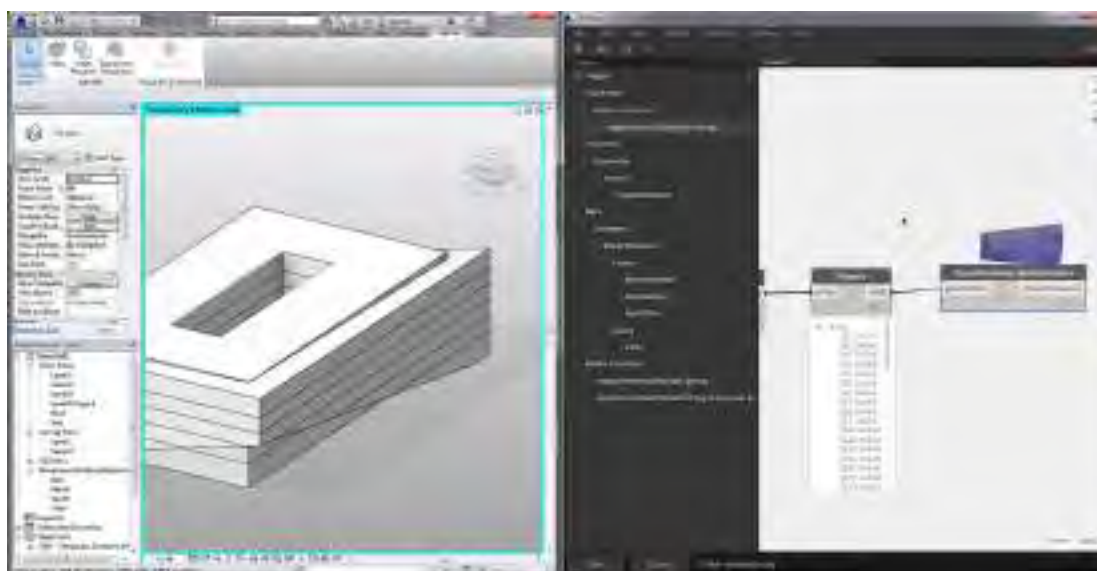
El uso convencional de Dynamo ha estado enfocado mayormente en aplicaciones relacionadas con la generación de arquitecturas complejas, análisis estructural y la creación de planos y documentos propios de los proyectos (Dynamo BIM, 2022; Monteiro, 2016). Sin embargo, es importante destacar que el acceso a esta herramienta dentro del software BIM Revit puede ser un tanto relegado y poco conocido, lo que limita su utilización por parte de la mayoría de los usuarios.

Algunos trabajos de investigación y guías proporcionadas por desarrolladores oficiales de Dynamo han resaltado su aplicación en la generación de formas arquitectónicas y el diseño estructural, como se puede observar en la Figura 4.1 (Dynamo BIM, 2022). Además, existen demostraciones y guías públicas disponibles que también enfatizan el uso de Dynamo en las especialidades de arquitectura y estructura, como se muestra en la Figura 4.2.

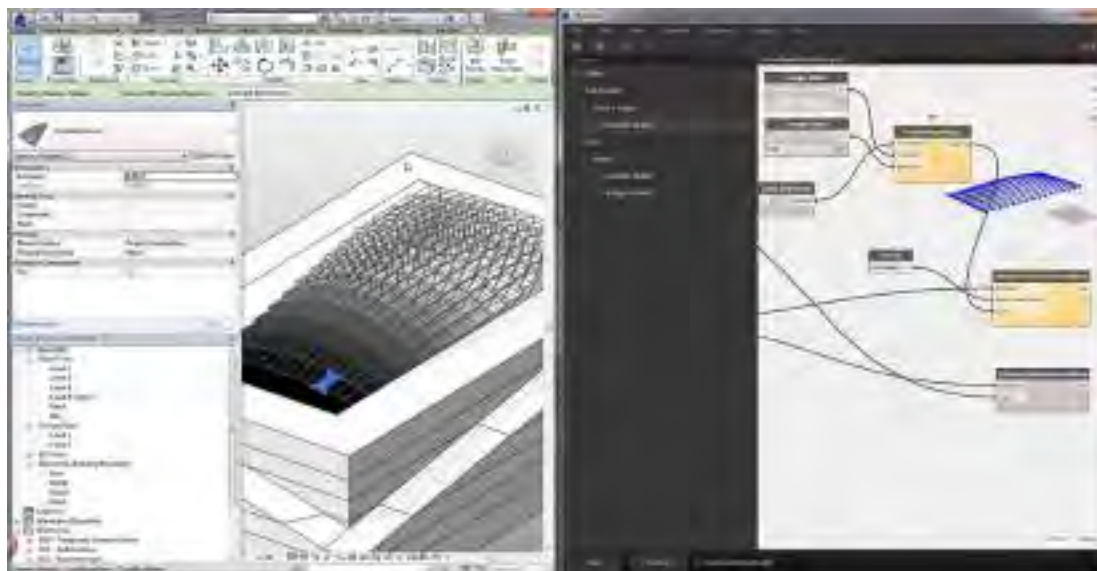
A pesar de que Dynamo ha demostrado su utilidad en estos campos, es importante reconocer que su potencial no se limita a estas especialidades. Existen oportunidades para explorar su aplicación en el diseño y cálculo de instalaciones eléctricas, aprovechando su capacidad paramétrica-visual para optimizar tareas repetitivas y tediosas, como las que se presentan en el modelado manual de instalaciones eléctricas en Revit.

**Figura 4.1**

*Operación de Dynamo con fines arquitectónicos y estructurales*



(a) *Dynamo operando volúmenes arquitectónicos.*



(b) *Dynamo generando estructuras.*

*Fuente: Capturas de software tomadas de Dynamo BIM (2022).*

## Figura 4.2

*Otros usos para Dynamo*



(a) *Dynamo generando arquitecturas complejas<sup>1</sup>*



(b) *Dynamo en diseño de estribos<sup>2</sup>*



(c) *Dynamo en diseño estructural de cubiertas<sup>3</sup>*

*Fuente:* Recopilación de capturas de software tomadas de demostraciones públicas en internet (\*1 Extracto del canal de “Domina Dynamo” <https://youtu.be/GeyHyDuptYY>, \*2 Extracto del canal de “Engineer Geek” <https://youtu.be/xg7DahD-tiU>, \*3 Extracto del canal de “Carlos Saravia” <https://youtu.be/qlksASTg7uM>

### 4.3.1 Estilo de Programación

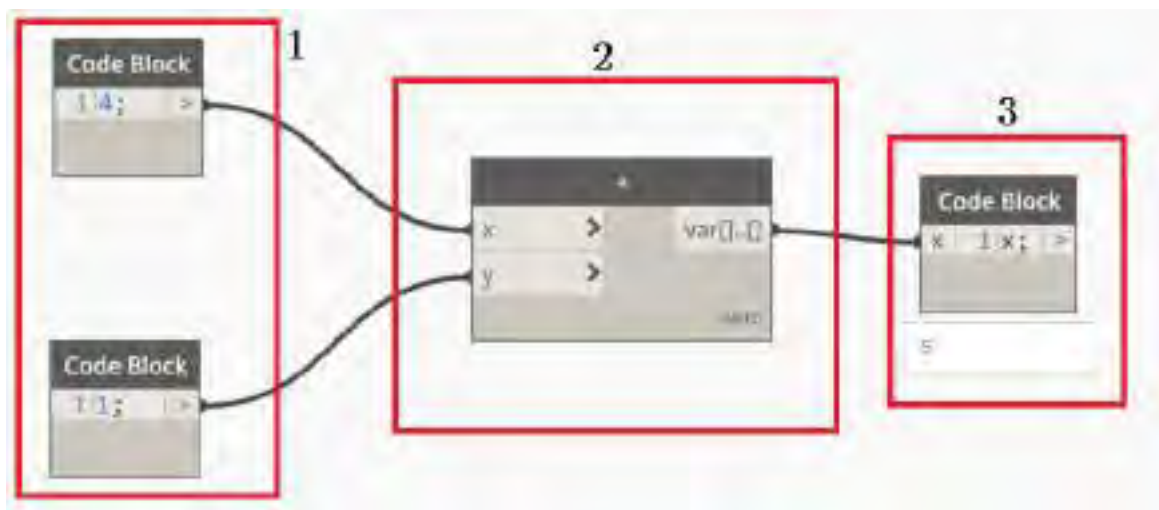
El acceso a Dynamo desde la interfaz de Revit permite una rápida interoperabilidad, lo que conduce a un estilo de programación que se basa en los elementos y parámetros presentes en el proyecto en el que se desee implementar la ejecución de scripts.

Al igual que Grasshopper, Dynamo emplea un enfoque de programación a través de “Nodos y Enlaces”. Los nodos funcionan mediante la definición y llamado de “funciones”, similar a la programación tradicional, lo que implica la necesidad de tener datos de entrada y salida.

En la Figura 4.3, se muestran tres elementos vitales dentro de este estilo de programación: los elementos enmarcados en **1** son los parámetros de entrada, **2** representa la función utilizada y **3** muestra la respuesta o resultado.

**Figura 4.3**

*Sistema de Nodos, Enlaces, Entradas y Respuestas de Dynamo*



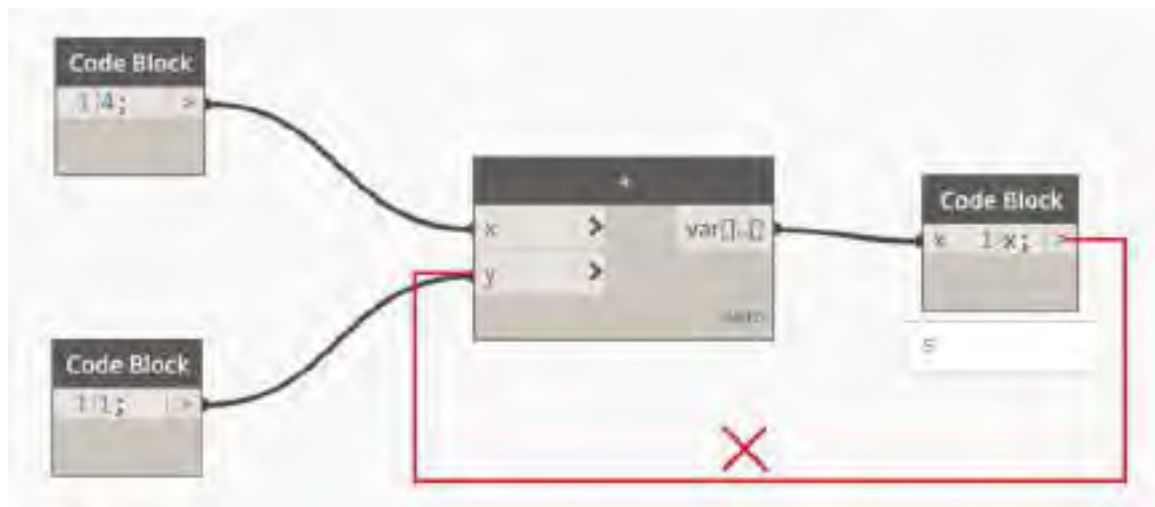
*Fuente:* Captura de software propia.

El sistema de nodos y enlaces asegura un flujo de programación consecuente y lineal, lo que impide retroceder resultados hacia las entradas de una misma función (ver Figura 4.4). Esto significa que la implementación de procesos recursivos se ve limitada dentro del sistema de Nodos y Enlaces de Dynamo. No obstante, esta limitación se puede superar aprovechando la capacidad de programación en Python, ya que Dynamo permite interactuar con este lenguaje de programación para realizar tareas más complejas y recursivas.

Este aspecto provoca que, como expuso Kirschner (2015) respecto a Grasshopper 3D (Ver pág. 56), en Dynamo también se encuentra limitado para la implementación de procesos recursivos dentro del sistema de Nodos y Enlaces; sin embargo, esta tarea no es imposible ya que la interfaz de Dynamo aprovecha la capacidad programable de Python (Dynamo BIM, 2022).

**Figura 4.4**

*Procesos recursivos en Dynamo*



*Fuente:* Captura de software propia.

La Figura 4.5 muestra un ejemplo básico de implementación de un proceso recursivo en Dynamo utilizando Python. A través de esta ventana hacia la interfaz de Python, se abre la posibilidad de incluir programas que manipulen con mayor precisión los parámetros presentes en el proyecto de instalaciones eléctricas interiores en el software BIM Revit. Esto requeriría aprender el código específico de Revit para lograr una mejor adaptación a las necesidades de los profesionales involucrados en el diseño y cálculo de instalaciones eléctricas.

### 4.3.2 *Requerimientos de Información*

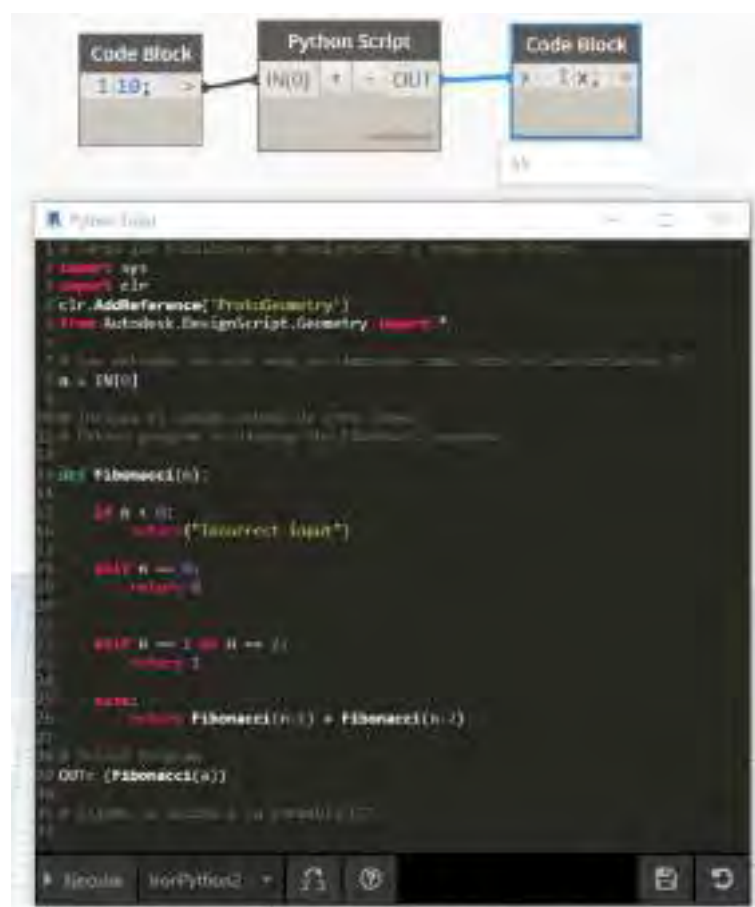
Dada la importancia de la característica informativa de los procesos BIM como metodología y también dentro del mismo software, Dynamo requiere dentro de su interfaz tener en cuenta la correcta diferencia entre *Familia*, *Tipo* y *Ejemplar*, tal como se muestra en la Figura 3.15. El manejo preciso de estos elementos es crucial para la manipulación de los parámetros de un objeto, desde sus propiedades únicas como elemento hasta su emplazamiento dentro del proyecto.

Gracias a esta manipulación, es posible automatizar procesos de cambio de parámetros para una gran cantidad de elementos sin demoras significativas. En el caso específico de las instalaciones eléctricas, la toma de parámetros como tensión, potencia, circuito y tablero (Figura 4.6) permite también automatizar procesos de cálculo.

Además, Dynamo ha ampliado su potencial al permitir la inclusión de nodos desarrollados por terceros a través de un repositorio oficial. Esta estrategia ha proporcionado mayor independencia

**Figura 4.5**

*Implementación de procesos recursivos en Dynamo*



*Fuente:* Implementación propia, adaptando el código fuente de GeeksforGeeks (2022) a Dynamo.

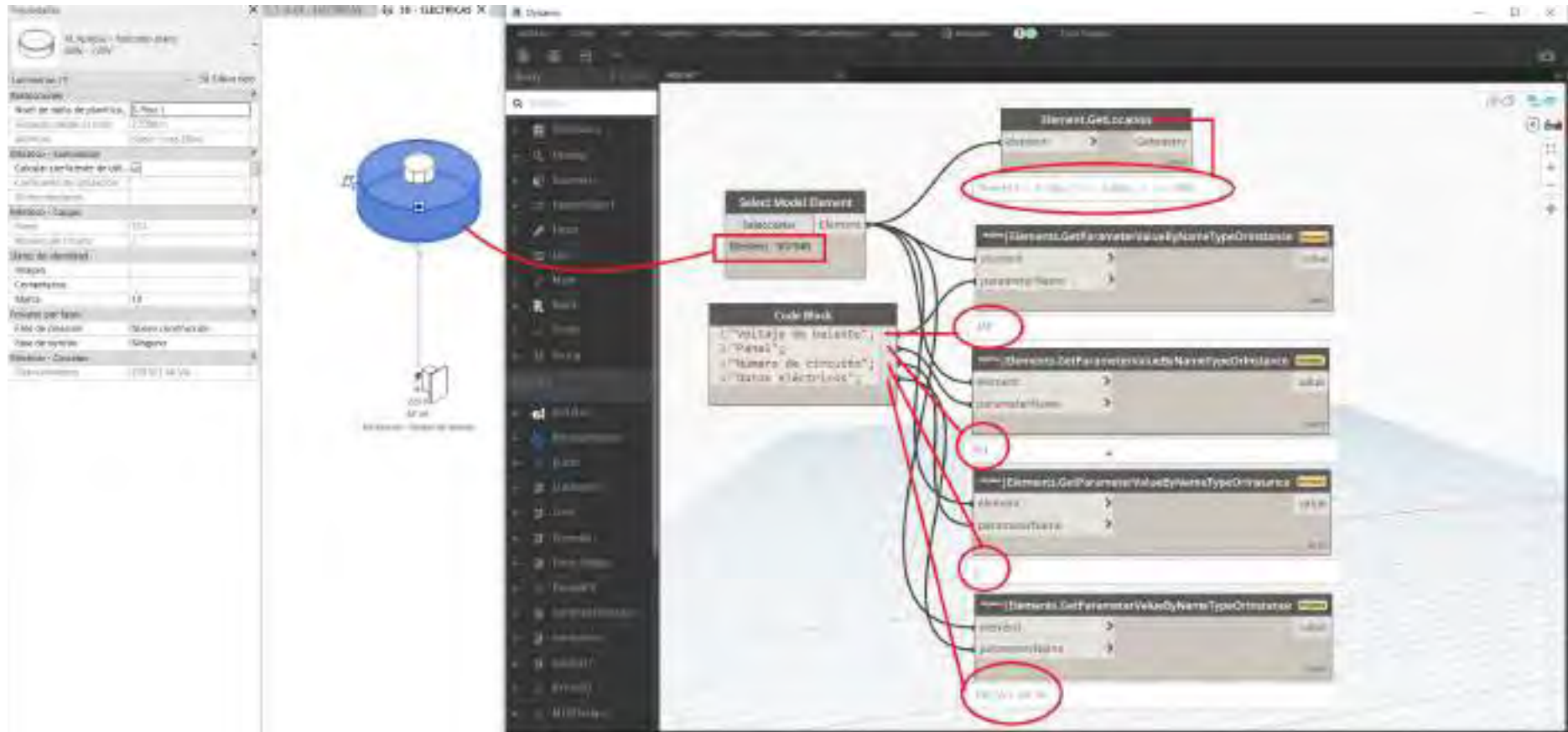
de sitios web y, al mismo tiempo, ha reducido el riesgo de contaminación informática por virus desarrollados en Python.

A pesar de contar con nodos predefinidos, esta inclusión de nodos externos ha simplificado la creación de nuevas funciones y ha disminuido la necesidad de recurrir a la programación directa en Python dentro de Dynamo.

Estas características de programación hacen que sea importante contar con conocimientos básicos de programación, ya que, cuando se desea manejar una gran cantidad de información, es necesario utilizar listas o vectores de elementos; la manipulación de estos es similar a enfoques tradicionales de programación.

Figura 4.6

Manejo de información de un elemento eléctrico en Dynamo



Fuente: Captura de software propia.

#### 4.4 Potenciales de la aplicación de Dynamo

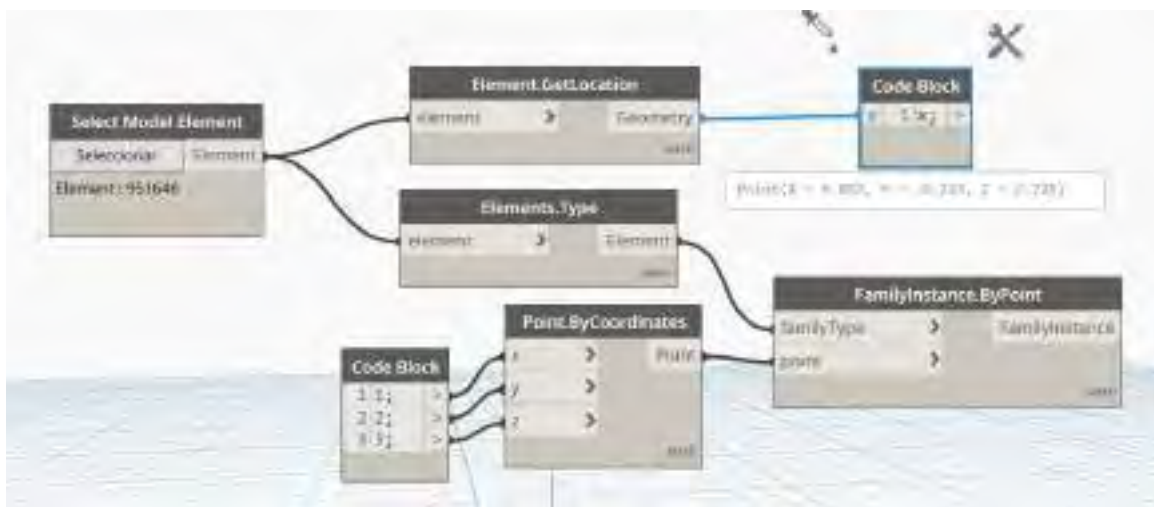
Ahora que se ha comprendido el funcionamiento de Dynamo en relación con el software BIM Revit y los elementos paramétricos que este posee, es posible identificar las posibilidades abiertas para la aplicación objetivo que presenta esta tesis. A continuación, se han diferenciado tres etapas importantes a considerar en la especialidad de instalaciones eléctricas interiores.

##### 4.4.1 Diseño

Dentro del campo de diseño de instalaciones eléctricas interiores, la principal consideración radica en aplicar criterios espaciales y de ubicación precisos para el proyecto. En este sentido, Dynamo ofrece grandes posibilidades para manipular coordenadas XYZ y traducirlas en puntos geométricos que permitan ubicar con precisión los equipos eléctricos en el espacio (Ver Figura 4.7).

**Figura 4.7**

*Ubicación y emplazamiento de elementos en Dynamo*



*Fuente:* Captura de software propia.

Además, Dynamo facilita la ubicación exacta de los puntos de conexión de los elementos eléctricos, lo que resulta especialmente útil en el diseño de rutas de conductos para alojar los conductores eléctricos (Ver Figura 4.8). Esto abre la puerta a la automatización del trazado de circuitos, agilizando significativamente el proceso de diseño.

La comprensión de elementos básicos de geometría como puntos de referencia para traducirlos en ubicaciones específicas de materiales y componentes eléctricos, brinda un amplio abanico de opciones para programar y automatizar el diseño de instalaciones eléctricas interiores.



Figura 4.8

Manipulación de conectores y extracción de información de un elemento en Dynamo



Fuente: Captura de software propia.

#### 4.4.2 Cálculo

Para el cálculo eléctrico dentro de un proyecto de instalaciones eléctricas interiores, sea uno simple o complejo, se consideran los principales cálculos:

1. Cálculo de cargas.
2. Cálculo de caídas de tensión.

En el proceso de cálculo de ingeniería, los cálculos de cargas y caídas de tensión pueden ser traducidos a tablas para su presentación y documentación. Con Dynamo, se puede aprovechar su capacidad extractiva de información para obtener datos relevantes, como área, tipo de edificación y potencia de las cargas, que son necesarios para el planteamiento de un cuadro de cargas. Sin embargo, es importante proporcionar a la interfaz los datos normativos provistos por el CNE–Utilización en la *Sección 050* para garantizar que los cálculos estén en conformidad con la normativa peruana, ya que Revit tiene limitaciones para realizar este tipo de cálculos, especialmente con cargas superficiales.

En cuanto al cálculo de caídas de tensión, Dynamo permite programar aspectos visuales que facilitarían la evaluación rápida de los puntos donde la tensión sobrepasa los límites establecidos. Esta aplicación encuentra su utilidad en proyectos muy grandes, donde el análisis de caídas de

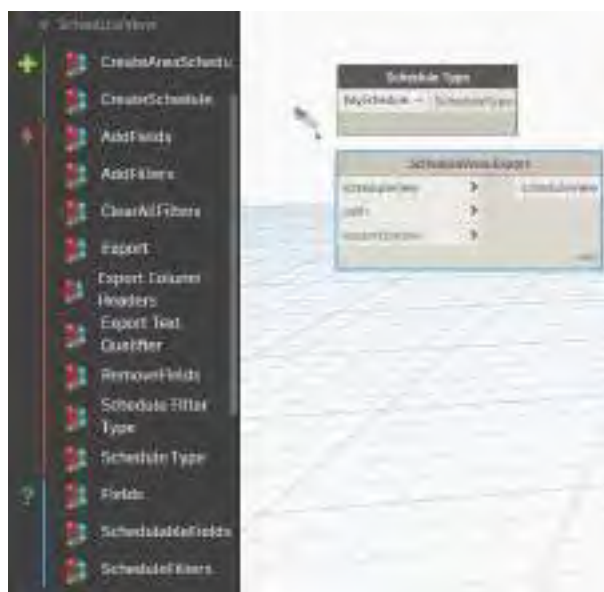
tensión puede ser más práctico a través de la observación que mediante el seguimiento de un cuadro de caídas de tensión. Pese a lo anterior, no se descarta la posibilidad de generación de un cuadro correspondiente.

Revit ofrece la funcionalidad adicional de exportar las tablas generadas en formatos como *.txt* o *.csv*. Aunque estos formatos son ampliamente utilizados para el procesamiento de datos en forma de texto, trabajar con ellos en *Microsoft Excel* puede presentar desafíos en ocasiones. Sin embargo, al utilizar Dynamo como intermediario, se pueden aprovechar las capacidades de integración y comunicación entre Revit y Excel.

Dynamo permite establecer una conexión directa y fluida entre Revit y Excel, lo que facilita una transferencia de datos más eficiente y precisa. A través de la programación de nodos específicos en la interfaz de programación de Dynamo, se puede configurar la exportación de los resultados directamente hacia Excel, lo que ofrece una mayor capacidad para realizar análisis detallados de los resultados obtenidos. Al utilizar Dynamo como intermediario, se mantiene la programación a niveles más simples y se evitan las complejidades propias de las hojas de datos.

#### Figura 4.9

*Nodos por defecto de Dynamo para la manipulación de tablas dentro de Revit*



*Fuente:* Captura de software propia.

Aunque la interoperabilidad entre Dynamo y Excel requiere un conocimiento mínimo de los parámetros informáticos del proyecto, como la ruta digital en la unidad de almacenamiento del computador o en el servidor que aloje el proyecto, esta integración amplía significativamente las posibilidades de manipulación y presentación de datos en proyectos de instalaciones eléctricas interiores.

En el caso de la manipulación de tablas en Revit, es importante mencionar que es una funcionalidad bastante cerrada y puede llevar a errores al intentar generar nuevas tablas del mismo tipo. Por tanto, las opciones para tareas que requieran una tabulación se limitan a la programación que contemple las tablas y datos existentes. Una alternativa más sencilla es recurrir a software de terceros, como la herramienta *DiRoots One*, desarrollada por la empresa portuguesa *DiRoots*, la cual es gratuita y presenta una buena interacción con el producto de Autodesk, especialmente en el ámbito de importación de datos externos hacia Revit.

#### **Figura 4.10**

*Diroots Pluggin BIM*



*Fuente:* Icono de software tomado de Diroots AEC Industry (2022)

#### **4.5 Marco Previo para el Desarrollo de Módulos**

El desarrollo de módulos para la automatización de diseño y cálculo de instalaciones eléctricas interiores debe considerar los siguientes aspectos:

1. Utilización de Dynamo: Dynamo se presenta como la mejor opción para interactuar con Revit debido a su compatibilidad directa y su capacidad de modificación, creación y manipulación paramétrica de elementos de un proyecto BIM.
2. Diseño de instalaciones eléctricas: El objetivo es automatizar el emplazamiento y direccionamiento del equipo eléctrico proyectado, permitiendo diseños personalizados de escenas de iluminación sin necesidad de renders arquitectónicos. Dynamo debe abarcar estas necesidades dentro de la interfaz de Revit o mediante interacciones con otros software especializados.
3. Cálculo de ingeniería: Se debe considerar la normativa peruana vigente para el cálculo de las instalaciones eléctricas. Es importante evitar el cálculo por defecto de Revit y desarrollar algoritmos que apliquen la normativa de manera precisa.

4. Interacción con Excel: Para el manejo de tablas y datos, se recomienda utilizar la integración entre Dynamo y Excel para una transferencia eficiente de información. Esto permitirá realizar análisis detallados de los resultados obtenidos.
5. Programación de módulos: Los algoritmos se representarán en forma de diagramas de flujo dentro de la interfaz visual de Dynamo. En caso de requerir programación adicional con Python, se pueden utilizar algoritmos en forma de pseudocódigo.

## CAPÍTULO V

### DESARROLLO DE MÓDULOS

#### 5.1 Introducción

El objetivo de este capítulo es plantear algoritmos que permitan la automatización de tareas de diseño y cálculo de instalaciones eléctricas interiores utilizando la operación directa entre Revit 2022 y Dynamo. Se busca aprovechar los potenciales de Dynamo para superar las limitaciones del software BIM y generar modelos de alta calidad que cumplan con las características BIM y las normativas vigentes.

Es importante entender que los algoritmos son la base de las herramientas informáticas que se desarrollan, y se traducen a un lenguaje de programación para su ejecución. En el contexto BIM, los flujos de trabajo pueden no ser consecutivos, por lo que se debe tener en cuenta que los ingenieros electricistas pueden utilizar la herramienta de manera flexible, siempre que se cumplan las características BIM y se sigan las referencias normativas correspondientes.

En este capítulo se detallan los procesos algorítmicos utilizando los nodos por defecto de Dynamo y también aquellos desarrollados por terceros, siempre que estén disponibles y funcionales. Además, se enfoca en el sustento normativo necesario para implementar los algoritmos en los programas, asegurando así que las herramientas cumplan con las regulaciones y estándares aplicables a las instalaciones eléctricas interiores.

El resultado de este capítulo serán herramientas automatizadas que permitan mejorar la eficiencia y precisión en el diseño y cálculo de instalaciones eléctricas interiores, al mismo tiempo que se promueve la integración de la metodología BIM en los procesos de ingeniería eléctrica.

#### 5.2 Recursos Adicionales

Para el proceso de programación y debido a la necesidad de interactuar con otros software, es importante considerar recursos adicionales que estén dentro del espacio de compatibilidad con Dynamo y, por consiguiente, con Revit. Estos recursos serán fundamentales para el desarrollo de las herramientas detalladas posteriormente. Se buscará utilizar software que permita un uso libre y eficiente en conjunto con Dynamo para ampliar las capacidades y funcionalidades en el diseño y cálculo de instalaciones eléctricas interiores.

### 5.3 Modelado

Dentro de las tareas de modelado tridimensional para efectos del BIM, el software BIM exige un apoyo en vistas auxiliares que puedan permitir manipular la ortogonalidad de los conductos eléctricos y en el enrutado de estos, como se mostró anteriormente en la Figura 3.13 (pág. 81). En este sentido, las herramientas que se deben programar deben centrarse en la generación automatizada de conductos sin necesidad de recurrir a múltiples vistas, sino más bien, aprovechar la tercera dimensión que el BIM provee para fines de alimentadores, iluminación, tomacorrientes o equipos de corriente débil.

Además, en el emplazamiento de unidades de carga o equipo eléctrico, se deben considerar no solo las ubicaciones de estos elementos, sino también los elementos que alojen el cableado o empalmes correspondientes, como cajas de paso, octogonales y/o rectangulares.

Sin embargo, es importante tener en cuenta que se debe brindar soporte para ubicaciones óptimas, especialmente cuando se trata de puntos de iluminación. Aunque Revit posee un conjunto de datos parametrizados con información sobre las propiedades lumínicas de cada elemento, el proceso de cálculo demuestra que no es del todo preciso y que le falta mucha interacción con el aspecto normativo. Por tanto, se debe proveer una herramienta que permita conectar provisionalmente a Revit con otro software que posibilite un mejor manejo de estas cualidades lumínicas.

Es relevante destacar que en todo momento se prioriza el uso de herramientas con licencias gratuitas o de tipo académico, ya que dentro del desarrollo académico de la presente tesis se busca promover la aplicación y mejora continua de estas herramientas por parte de la comunidad académica interesada.

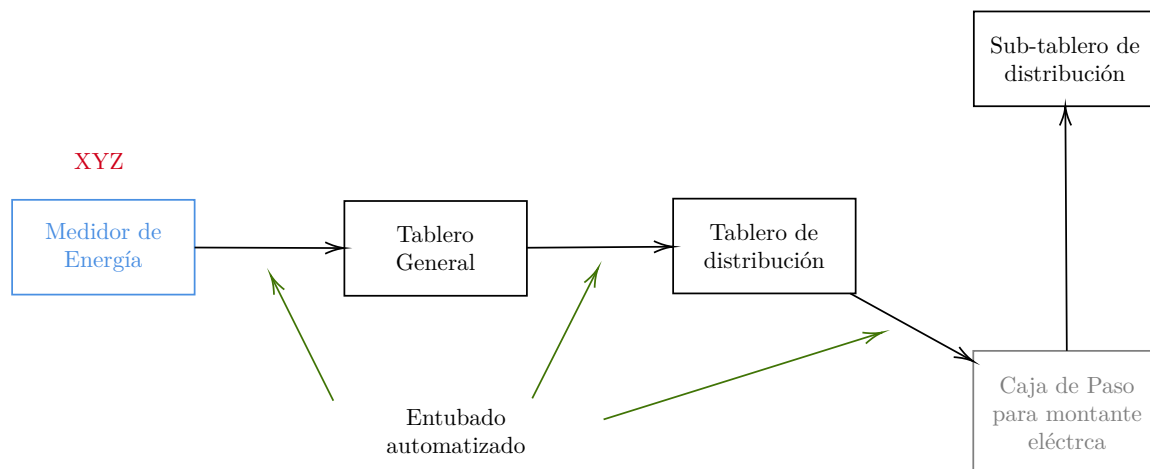
#### 5.3.1 *El Boceto Eléctrico a Programar*

En esta primera meta programable, se desea desarrollar una herramienta que permita generar modelos BIM correspondientes a conductos eléctricos, como por ejemplo, tuberías PVC en sus clases Pesada y Liviana, ya que este material es el más utilizado en procesos constructivos. El programa debe ser de código abierto y estar en constante mejora, lo que permitirá ampliar su funcionalidad para incluir más tipos de conductos en el futuro.

Para lograr esto, el programa requerirá que elementos como medidores de energía, tableros o cajas de paso estén previamente ubicados en el modelo BIM, y que la ubicación de estos elementos sirva como datos de entrada para el programa. Esto permitirá que el programa pueda obtener información sobre la ubicación y nivel de estos elementos, lo cual es importante para la generación de montantes eléctricas, como se muestra en la Figura 5.1.

**Figura 5.1**

*Boceto eléctrico a programar N°1 – Alimentación*

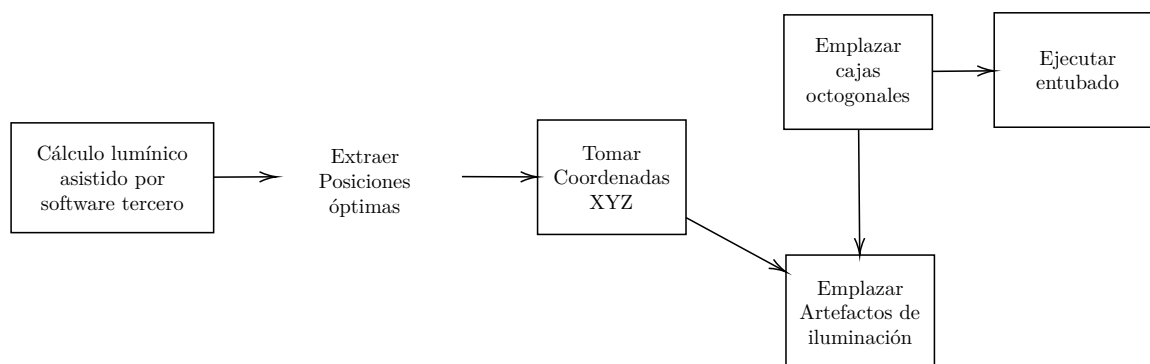


*Fuente:* Elaboración propia.

Además, dentro del mismo campo funcional, se busca lograr la interoperatividad con software de cálculo lumínico que pueda ser configurado según la normativa aplicable. De esta manera, se podrán ubicar de forma automatizada los artefactos de iluminación, como se muestra en la Figura 5.2, basándose en un estudio previo que se haya realizado. Esta funcionalidad permitirá estudiar el modelo BIM desde las perspectivas de Necesidad, Información y Gestión.

**Figura 5.2**

*Boceto eléctrico a programar N°2 – Iluminación*



*Fuente:* Elaboración propia.

Para las demás tareas, se aplicarán lógicas similares en busca de reducir el esfuerzo computacional necesario para ejecutar el software BIM en su totalidad.

### 5.3.2 *Criterio profesional inicial*

Dadas las delimitaciones establecidas en el primer capítulo de la presente tesis, no se pretende desarrollar una inteligencia artificial que pueda ubicar automáticamente los puntos iniciales de modelado, como medidores o tableros eléctricos. Es importante destacar que los programas desarrollados dependerán completamente del criterio profesional del ingeniero electricista y requerirán una interacción manual entre el usuario y el software BIM para su funcionamiento básico. Esto significa que el ingeniero electricista deberá proporcionar la ubicación y otros datos necesarios para que el programa pueda generar los modelos BIM correspondientes de manera adecuada.

### 5.3.3 *Recursos Adicionales*

Para el desarrollo de esta sección programable enfocada al modelado 3D, se considera el apoyo de recursos adicionales que sean compatibles con Revit y Dynamo. Estos recursos permitirán ampliar las capacidades de los programas desarrollados y facilitarán la interacción con otras herramientas y formatos.

**5.3.3.0.1 Dialux EVO 10.1** Dialux EVO 10.1 es una herramienta informática diseñada para realizar diseños de iluminación y cálculos lumínicos. Su interfaz sencilla y su apartado normativo facilitan al ingeniero electricista la verificación de diseños lumínicos personalizados. La elección de Dialux EVO se debe a su licencia libre de uso y su compatibilidad con formatos BIM como el *.ifc*, así como su capacidad para exportar datos en formatos *.dwg* o CAD.

#### **Figura 5.3**

*Dialux EVO 10.1 Software*



*Fuente:* Icono de software tomado de DIAL GmbH (2022)

**5.3.3.0.2 AutoCAD 2022** AutoCAD 2022 es un software CAD que será utilizado en esta ocasión para el manejo de archivos de dibujo e información en formato *.dwg*. Este recurso permitirá



obtener coordenadas y exportar información hacia formatos compatibles con Dynamo, en línea con los requerimientos del boceto eléctrico N°2 (Figura 5.2).

#### **Figura 5.4**

*AutoCAD 2022 Software*



*Fuente:* Icono de software tomado de Autodesk Latinoamérica (2022a)

**5.3.3.0.3 Archivos de texto básico .txt** Los archivos de texto en formato *.txt* serán utilizados para el procesamiento de cadenas de texto dentro de las interfaces de programación. Este formato permite almacenar información textual sin formato alguno y será de gran utilidad para la compatibilidad de lectura de cadenas de texto entre Dynamo y AutoCAD, ya que este último puede exportar información en este formato.

#### **5.3.4 Alimentación Eléctrica**

En esta primera fase de diseño, se plantea la programación de algoritmos basados en el criterio profesional inicial. En el flujo de trabajo BIM, el diseño de instalaciones eléctricas comienza después de un estudio de necesidades.

El enfoque principal se centra en los alimentadores eléctricos que parten del medidor. En el modelado tridimensional, la complejidad se presenta en las tareas de entubado, por lo que el primer programa a desarrollar se dedica a automatizar esta tarea.

El objetivo principal de este programa será agilizar y simplificar el proceso de entubado de alimentadores eléctricos, proporcionando una herramienta que pueda generar automáticamente las tuberías y conductos eléctricos necesarios para la conexión entre los elementos principales del sistema.

Con esta automatización, se espera mejorar la eficiencia en el diseño de instalaciones eléctricas y permitir que el ingeniero electricista se enfoque en aspectos más complejos y estratégicos del proyecto.

### 5.3.4.1 Base del Algoritmo

La base algorítmica para la herramienta de entubado de alimentadores eléctricos se enfoca en el diseño de instalaciones eléctricas dentro de un flujo de trabajo BIM. La entrada principal son los elementos esenciales como tableros y cajas de paso, que contienen puntos de conectores con ubicaciones específicas basadas en información BIM.

Uno de los desafíos programables radica en la identificación de conectores ubicados en diferentes caras de los elementos eléctricos, es decir, si se encuentran en caras superiores, inferiores o laterales. Es fundamental tener en cuenta que, en aplicaciones reales, un conector no puede ser utilizado por dos tubos a la vez, lo que implica que cada conector utilizado debe excluirse de las opciones restantes.

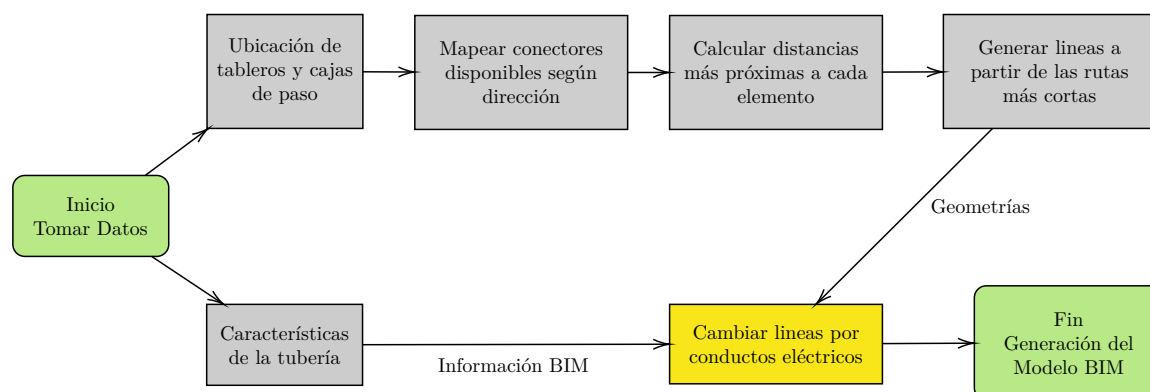
Además, la operación estándar de Dynamo permite seleccionar varios elementos a la vez, pero para esta herramienta, se requerirá algún proceso repetitivo o recursivo para abordar cada elemento individualmente y realizar las conexiones necesarias.

Otro aspecto importante es la priorización del entubado de la tubería eléctrica, dándole preferencia a los conectores más cercanos al elemento de inicio. Esto se traduce en una mayor eficiencia y coherencia en el diseño de la instalación eléctrica.

Por último, los elementos como tuberías, pueden ser emplazados según geometrías básicas como punto, línea o plano. Por tanto, es necesario utilizar estas geometrías para traducir las conexiones entre los elementos eléctricos en líneas que representen las tuberías eléctricas.

**Figura 5.5**

*Diagrama de Flujo para el script “Conductos-i”*



*Fuente:* Elaboración propia.

La Figura 5.5 muestra un diagrama de flujo que aproxima la tarea de programación para el desarrollo del programa “Conductos-i”. Se prevé que la implementación requerirá programación

tradicional dentro de la interfaz de Dynamo para lograr procesos repetitivos que permitan la toma masiva de datos de entrada y, posteriormente, la generación de las líneas de entubado.

En la implementación del programa, el Algoritmo 5.1 cumple su función de asignación de variables y llama a la función denominada *Lin2*, que se encuentra detallada en el Algoritmo 5.2.

Para desarrollar estos algoritmos, se propone aprovechar las entradas que ofrece Dynamo, ya que mantienen sus propiedades informativas y geométricas.

El objetivo principal de este primer agregado en programación tradicional es basar el cálculo matemático que se encuentra en tópicos básicos de geometría euclidiana para el cálculo de la distancia entre dos puntos en el espacio (Ecuación 5.2), esta ecuación es implementada en la línea 37 del Algoritmo 5.2.

$$D_{p1-p2} = \sqrt{(X_{p2} - X_{p1})^2 + (Y_{p2} - Y_{p1})^2 + (Z_{p2} - Z_{p1})^2} \quad (5.2)$$

Donde:

$D_{p1-p2}$	:	Distancia entre dos puntos en el espacio
$X_{p1}, Y_{p1}, Z_{p1}$	:	Coordenadas X, Y & Z del primer punto
$X_{p2}, Y_{p2}, Z_{p2}$	:	Coordenadas X, Y & Z del segundo punto

---

**Algoritmo 5.1** Cálculo del conector más próximo y generación de líneas

---

**Require:** Niveles (Lista), Puntos Conectores Superiores e Inferiores (Lista)

```

1: Lev ← Niveles
2: Sup ← Conectores Superiores por elemento
3: Inf ← Conectores Inferiores por elemento
4: TamEle ← Tamaño de lista Lev
5:  $X_{sup}, Y_{sup}, Z_{sup}$  ← Coordenadas (X, Y, Z) de Sup
6:  $X_{inf}, Y_{inf}, Z_{inf}$  ← Coordenadas (X, Y, Z) de Inf
7: aux ← []
8: LIN ← lista de ceros de tamaño de TamLev
9: USED ← lista de ceros de tamaño de TamLev
10:  $X_o, Y_o, Z_o$  ← 0
11:  $X_{ff}, Y_{ff}, Z_{ff}$  ← 0
12: for  $i = 0 \dots (TamEle - 1)$  do
13:   if  $Lev[i] = Lev[i + 1]$  then
14:      $LIN[i] \leftarrow Lin2(i, -1, i + 1, -1)$ 
15:   else
16:      $LIN[i] \leftarrow Lin2(i, 1, i + 1, -1)$ 
17:   end if
18: end for
   return LIN

```

*Fuente:* Elaboración propia.

---

En el desarrollo de la obra, los conductos o tuberías eléctricas no solo se extienden en horizontal, sino que también presentan una componente vertical debido a su altura de montaje o a su trayectoria ascendente hacia niveles superiores. Por lo tanto, es necesario completar el proceso de

generación de líneas que considere estas características verticales. Esto implica llevar la verticalidad de los conductos hacia los puntos donde están diseñadas las losas o pisos en el modelo.

El Algoritmo 5.3 describe una traslación espacial de puntos, y toma en cuenta dos casos básicos para dos elementos: si estos se encuentran en el mismo nivel o en diferentes niveles. Para esto, se han previsto dos funciones que se detallan en el Algoritmo 5.4. Es importante mencionar que esta explicación es relevante debido a que cada línea generada por los algoritmos 5.1 y 5.2 posee un punto de inicio y un punto final. Considerando el sistema tridimensional de ejes, esta operación se realiza principalmente en el eje  $Z$ .

Con los algoritmos previamente propuestos, es posible iniciar el desarrollo programable en Dynamo, ya que las herramientas necesarias para las etapas que no requieren el uso de bloques repetitivos dentro del diagrama de flujo, como se muestra en la Figura 5.5, se pueden configurar mediante nodos enlazables dentro de la interfaz paramétrica-visual que ofrece Dynamo.

---

**Algoritmo 5.2** Función *Lin2*, parte del Algoritmo 5.1

**Require:** listas: *Sup, Inf, X<sub>sup</sub>, Y<sub>sup</sub>, Z<sub>sup</sub>, X<sub>inf</sub>, Y<sub>inf</sub>, Z<sub>inf</sub>, USED*

```

1: function LIN2(a, da, b, db)
2:   Min  $\leftarrow$  10000
3:   M  $\leftarrow$  Vector inicializador vacío
4:   if da = 1 then
5:     Pi  $\leftarrow$  Sup[a]
6:     h  $\leftarrow$  Tamaño de la lista Sup[a]
7:     Xo, Yo, Zo  $\leftarrow$  Xsup[a], Ysup[a], Zsup[a]
8:   else if da = -1 then
9:     Pi  $\leftarrow$  Inf[a]
10:    h  $\leftarrow$  Tamaño de la lista Inf[a]
11:    Xo, Yo, Zo  $\leftarrow$  Xinf[a], Yinf[a], Zinf[a]
12:   else if db = 1 then
13:     Pf  $\leftarrow$  Sup[b]
14:     w  $\leftarrow$  Tamaño de la lista Sup[b]
15:     Xff, Yff, Zff  $\leftarrow$  Xsup[b], Ysup[b], Zsup[b]
16:   else if db = -1 then
17:     Pf  $\leftarrow$  Inf[b]
18:     w  $\leftarrow$  Tamaño de la lista Inf[b]
19:     Xff, Yff, Zff  $\leftarrow$  Xinf[b], Yinf[b], Zinf[b]
20:   end if
21:   for m = 1 ... Tamaño de lista Pi do
22:     if USED[a]  $\neq$  0 then
23:       if USED[a]  $\cong$  Pi[m] es Verdadero then
24:         Extender lista M con Pi[m]
25:         Quitar Xo[m], Yo[m], Zo[m]
26:       end if
27:     end if
28:   end for
29:   for m = 1 ... Tamaño de lista M do
30:     Quitar M[m] de lista Pi
31:   end for
32:   line  $\leftarrow$  0
33:   for i = 1 ... Tamaño de lista Pi do
34:     for j = 1 ... Tamaño de lista Pf do
35:       Xi, Yi, Zi  $\leftarrow$  Xo[i], Yo[i], Zo[i]
36:       Xf, Yf, Zf  $\leftarrow$  Xff[j], Yff[j], Zff[j]
37:       DIS  $\leftarrow$   $\sqrt{(X_f - X_o)^2 + (Y_f - Y_o)^2 + (Z_f - Z_o)^2}$ 
38:       if DIS < Min then
39:         line  $\leftarrow$  Generar línea entre Pi[i] y Pf[j]
40:         USED[b]  $\leftarrow$  Pf[j]
41:         Min  $\leftarrow$  DIS
42:       end if
43:     end for
44:   end for
45:   return line
46: end function

```

*Fuente:* Elaboración propia.

---

---

**Algoritmo 5.3** Traducción en eje  $Z$  para los puntos inicial y final de  $LIN$ 


---

**Require:** Coordenadas lineas almacenadas en  $LIN$ .

```

1:  $Z_i \leftarrow$  Coordenadas  $Z$  de puntos iniciales de  $LIN$ 
2:  $Z_f \leftarrow$  Coordenadas  $Z$  de puntos finales de  $LIN$ 
3:  $Lev \leftarrow$  Niveles
4:  $ELev \leftarrow$  Elevación de los Niveles
5:  $Z_{na} \leftarrow 0$ 
6:  $Z_{nb} \leftarrow 0$ 
7:  $Tam \leftarrow$  Tamaño de la lista  $Lev$ 
8:  $Z_N \leftarrow (Tam - 1)$  listas con dos 0 cada una
9: for  $i = 1 \dots (Tam - 1)$  do
10:   if  $Lev[i] = Lev[i + 1]$  then
11:      $TransInf(i)$ 
12:   else
13:      $TransSup(i, i + 1)$ 
14:   end if
15: end for

```

*Fuente:* Elaboración propia.

---



---

**Algoritmo 5.4** Funciones  $TransInf$  y  $TransSup$ , parte del Algoritmo 5.3

---

**Require:** listas:  $Z_i, Z_f, ELev$

```

1: function TRANSINF( $a$ )
2:    $Z_{na} \leftarrow -Z_i[a] + ELev[a]$ 
3:    $Z_{nb} \leftarrow -Z_f[a] + ELev[a]$ 
4:    $Z_N[a] \leftarrow [Z_{na}, Z_{nb}]$ 
5: end function
6: function TRANSUP( $a, b$ )
7:   if  $ELev[a] < ELev[b]$  then
8:      $Z_{na} \leftarrow -Z_i[a] + ELev[b]$ 
9:      $Z_{nb} \leftarrow -Z_f[a] + ELev[b]$ 
10:     $Z_N[a] \leftarrow [Z_{na}, Z_{nb}]$ 
11:   else
12:      $Z_{na} \leftarrow -Z_i[a] + ELev[a]$ 
13:      $Z_{nb} \leftarrow -Z_f[a] + ELev[a]$ 
14:      $Z_N[a] \leftarrow [Z_{na}, Z_{nb}]$ 
15:   end if
16: end function

```

*Fuente:* Elaboración propia.

---

### 5.3.4.2 Desarrollo del Script “Conductos-i”

El desarrollo del script “Conductos-i” requiere el uso de paquetes adicionales en Dynamo, ya que en el entorno “Electrical” de Revit no se encuentran nodos por defecto que faciliten la manipulación de elementos básicos como conectores, tuberías y curvas. A continuación, se enumeran los paquetes adicionales necesarios:

- Paquetes Adicionales:
  1. *Springs Nodes* por dimitar.ven<sup>1</sup>.
  2. *Gorilla17* por Gytaco<sup>2</sup>
  3. *MEPover* por tpover<sup>2</sup>
  4. *Data-Shapes* por Elayoubi Mostafa<sup>3</sup>

La toma de datos en el entorno de Revit es un paso crucial para que los Algoritmos 5.1 y 5.3 puedan funcionar correctamente. Sin embargo, Dynamo, la herramienta de programación visual utilizada para interactuar con Revit, presenta un desafío en cuanto a la selección de datos, ya que puede reordenarlos según distintos criterios como tipo, familia u otros. En este contexto, el uso de nodos de terceros se justifica aún más, ya que estos nodos adicionales ofrecen funcionalidades que facilitan la selección y manipulación de datos en el entorno de Revit; en ese sentido la Figura 5.6 muestra el medio nodal para la correcta selección de datos necesarios.

El desafío principal radica en mapear los conectores según su dirección, lo cual se puede abordar utilizando el manejo vectorial de la componente en el eje  $Z$  dentro de la programación de Dynamo. En este contexto, se pueden definir vectores para representar las direcciones de los conectores. Por ejemplo:

$$\begin{aligned} \vec{A}Conector &= (0, 0, +1) && \text{Dirección Superior} \\ \vec{A}Conector &= (0, 0, -1) && \text{Dirección Inferior} \end{aligned} \tag{5.3}$$

Estos vectores permiten identificar la orientación de cada conector en el eje  $Z$ , lo que será útil para determinar cómo se conectarán los elementos eléctricos en el modelo tridimensional.

En Dynamo, el proceso de mapeado mediante el manejo vectorial ofrece muchas posibilidades adicionales. En el caso específico del entubado de alimentadores eléctricos, cada elemento seleccionado tiene conectores con información matemática vital que permite una manipulación ágil y precisa.

La Figura 5.7 muestra cómo se realiza el mapeado de los conectores para cada elemento seleccionado. Cada conector está representado por una flecha que indica su dirección en el espacio

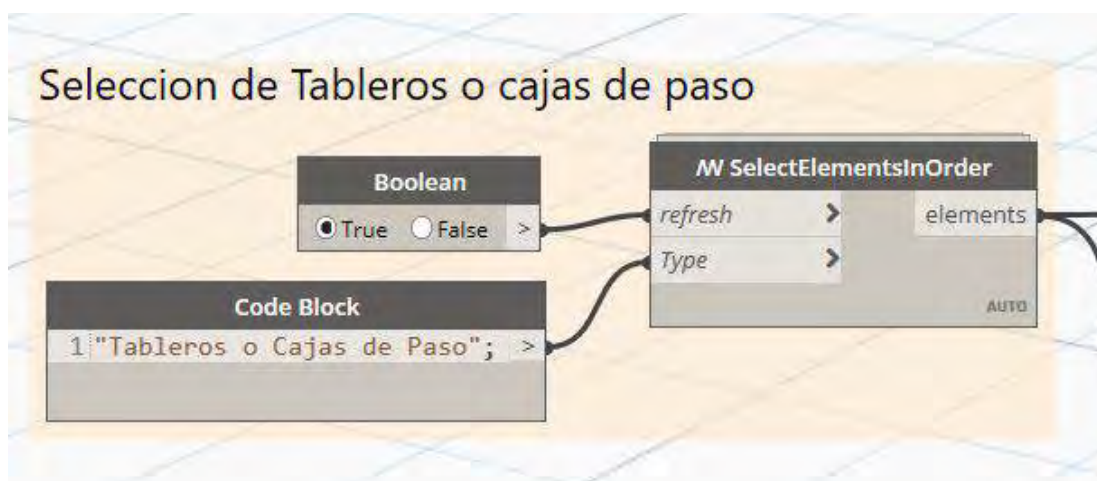
<sup>1</sup> Repositorio del autor: <https://github.com/dimven/SpringNodes>

<sup>2</sup> Consulta en Repositorio oficial de Dynamo: <https://www.dynamopackages.com/>

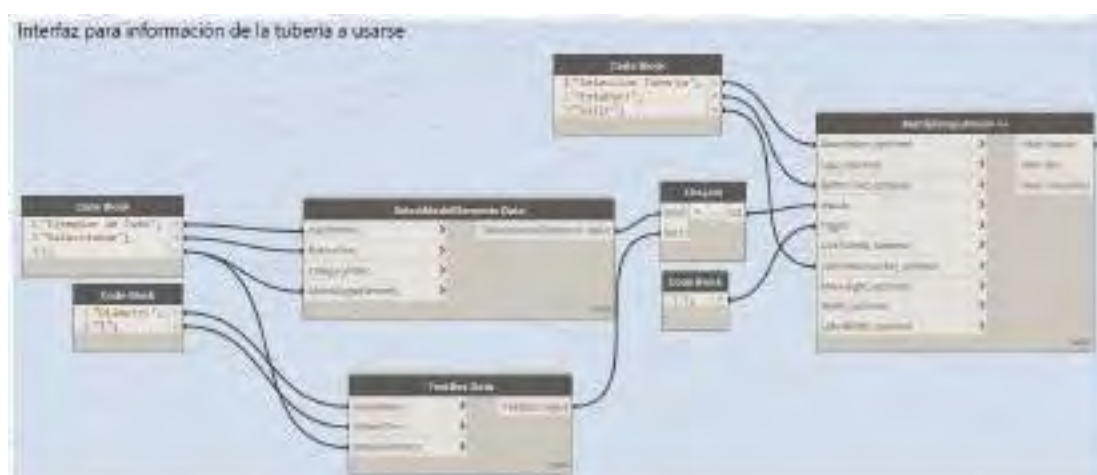
<sup>3</sup> Sitio Web del autor: <https://data-shapes.io/>

Figura 5.6

Interfaz para la selección de datos.



(a) Interfaz para la selección de cajas o tableros.



(b) Interfaz para la selección de tipo de tubo eléctrico.

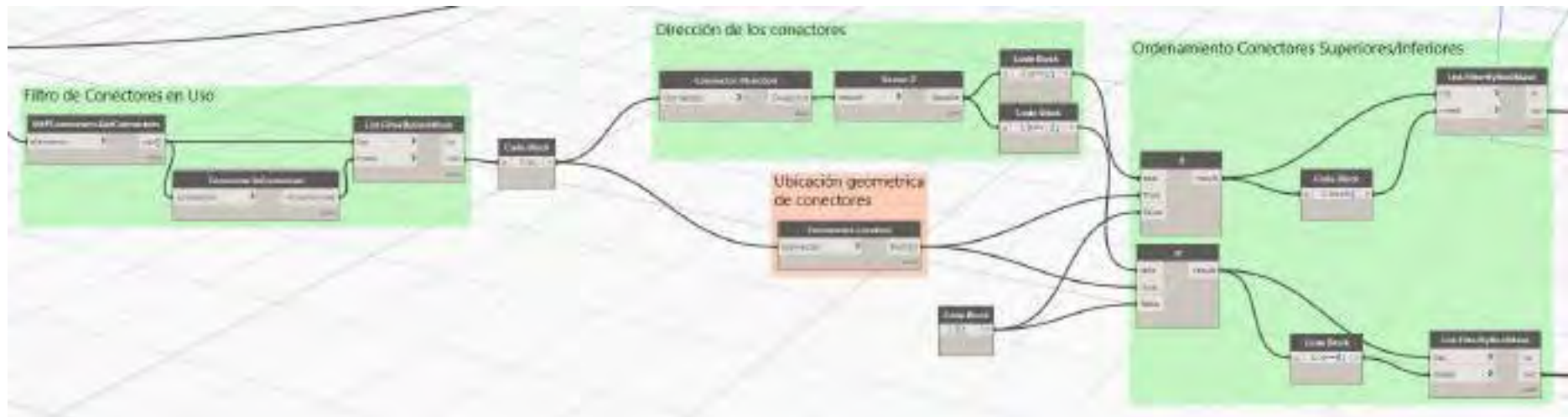
Fuente: Implementación propia.

tridimensional. Los conectores con dirección superior se representan con flechas hacia arriba, mientras que los conectores con dirección inferior se representan con flechas hacia abajo.



**Figura 5.7**

*Mapeado de conectores por elemento seleccionado*



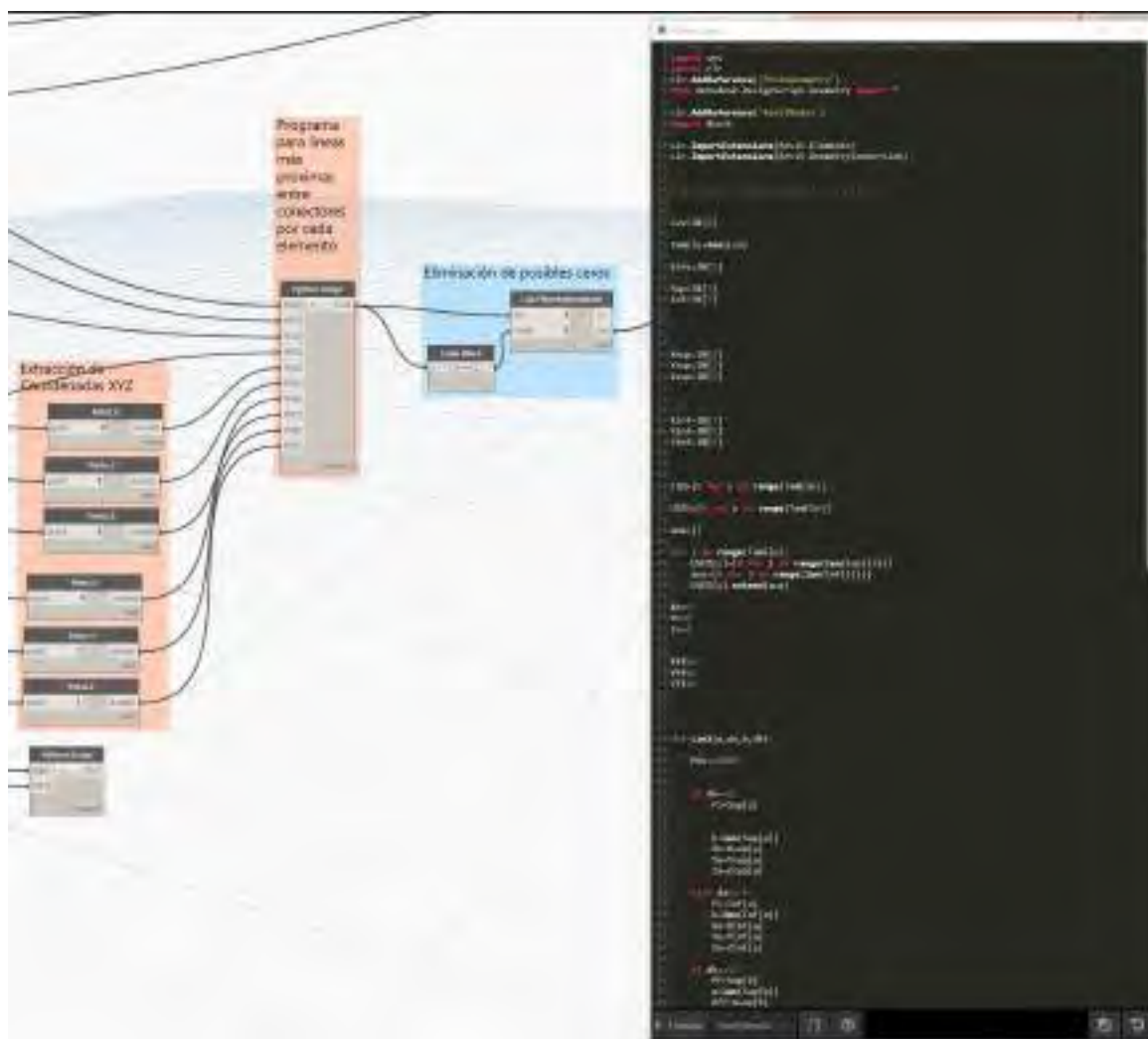
*Fuente:* Implementación propia.

La continuación del diagrama de flujo propone el cálculo de las distancias más cercanas entre elementos, así como la generación de líneas que permitan el modelado de los conductos eléctricos basados en estas distancias. Para lograrlo, se requieren las coordenadas base de los conectores superiores e inferiores, lo que lleva a utilizar la programación tradicional.

En la Figura 5.8, se muestra la implementación del Algoritmo 5.1 en Dynamo utilizando el lenguaje de programación Python. Este código permite calcular las líneas que conectarán los conectores de forma óptima, priorizando la distancia más corta entre ellos.

### Figura 5.8

*Implementación del Algoritmo 5.1 en Dynamo.*



*Fuente:* Implementación propia.

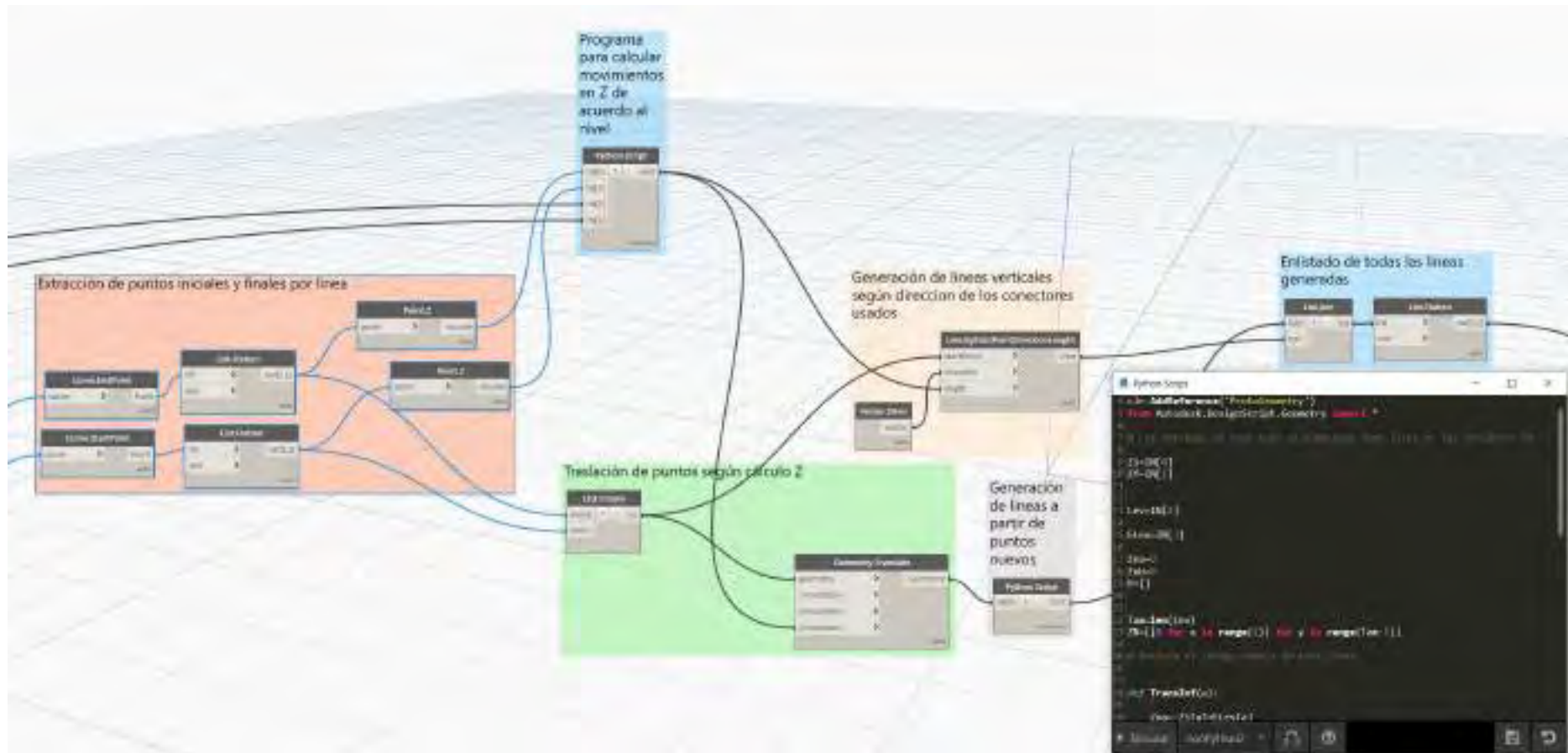
El código completo en Python se puede consultar en el Anexo A (Código A.1), donde se detalla cada paso del algoritmo y se muestra cómo se utiliza la información de los conectores para generar las líneas que representarán los conductos eléctricos en el modelo tridimensional.

A continuación, se realiza el tratamiento y la manipulación geométrica mencionada en la página 113, con el objetivo de traducir las montantes de subida y bajada de los tubos eléctricos. Para esto, se utiliza el algoritmo Algoritmo 5.3, cuyo código completo se puede consultar en el Anexo A (Código A.2).

El punto de partida para esta manipulación geométrica es la inicialización con las líneas generadas entre los conectores, considerando que se han hallado las distancias más cortas entre ellos, como se muestra en la Figura 5.9.

Figura 5.9

Traslación en ejes Z como traducción de montantes eléctricas.



Fuente: Implementación propia.

**Figura 5.10**

*Generación de conductos eléctricos basados en líneas geométricas.*



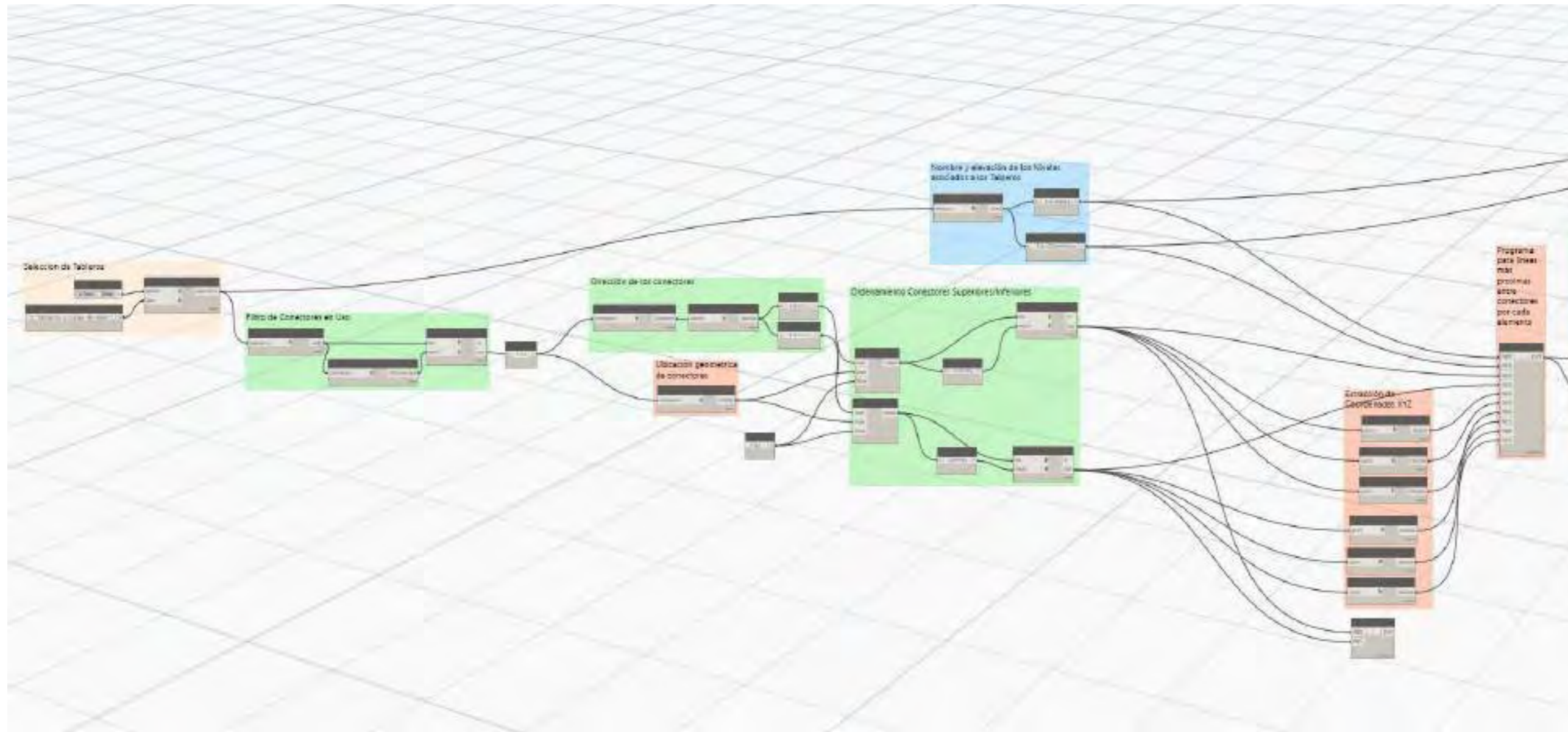
*Fuente:* Implementación propia.

La Figura 5.10 muestra el último proceso del diagrama de flujo, que consiste en el emplazado de los conductos eléctricos en base a las líneas generadas anteriormente. En este paso, se agregan en los lugares correspondientes las curvas o conectores que la Familia a emplazar ya tenga preconfiguradas. La información BIM de cada elemento se utiliza para diferenciar el tipo en la jerarquía de elementos, ya que solo los tipos proporcionan información más precisa en comparación con niveles más superiores para los elementos.

Los datos, como el diámetro a usarse en los conductos eléctricos, responden a la capacidad de manejo del entorno *Electrical* de unidades y las configuraciones previas establecidas. Asimismo, al mover los elementos sin perder información dentro de Revit, la profundidad de empotramiento dependerá de la capacidad de *Code Checking* y eventual *Clash Detection* que posea el usuario.

Figura 5.11

Vista completa del script Conductos-i

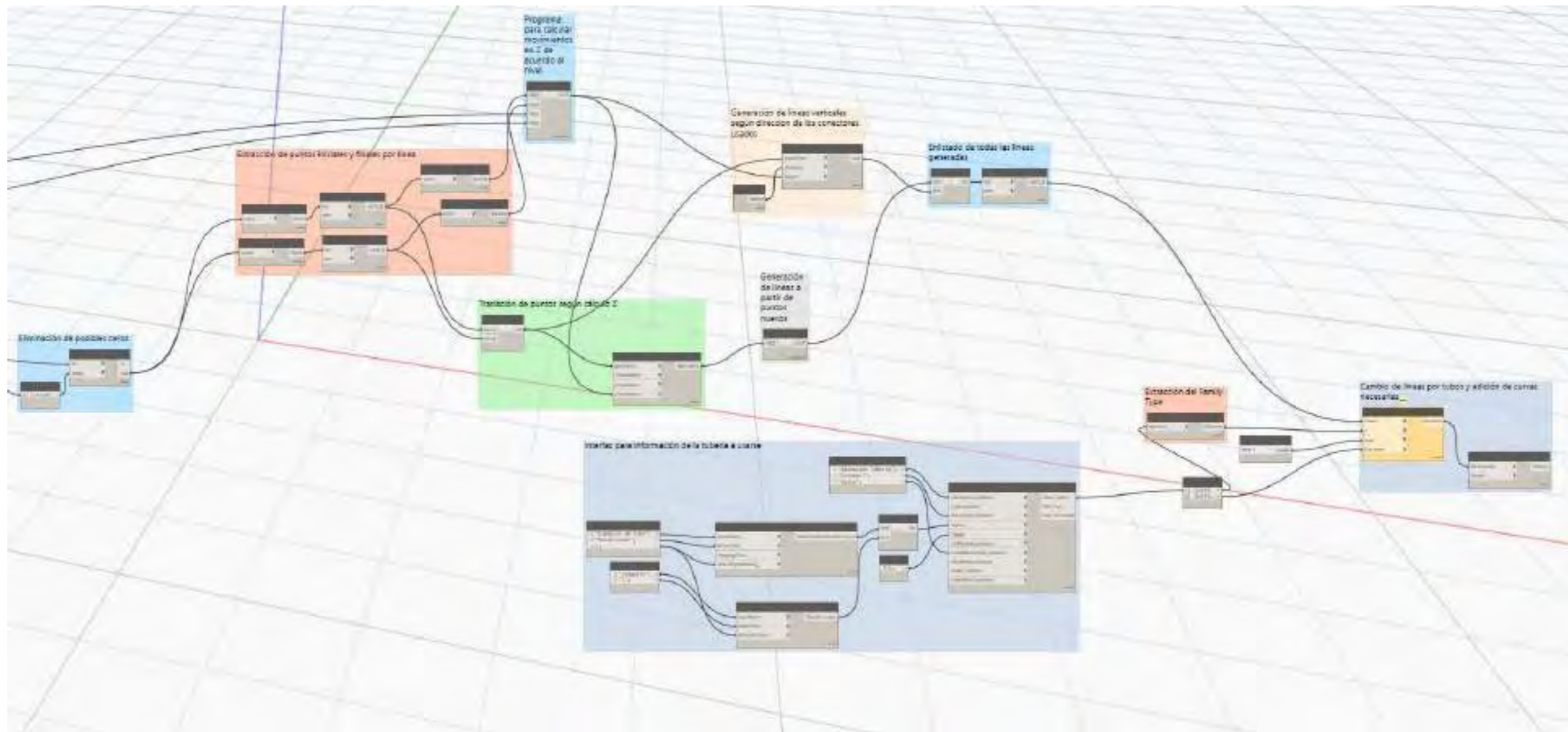


(a) Primera parte del script.

Fuente: Implementación propia.

Figura 5.11

Vista completa del script Conductos-i (Continuación)



(b) Segunda parte del script.

Fuente: Implementación propia.



### 5.3.5 Iluminación

Dentro de los proyectos de instalaciones eléctricas, tanto en enfoques interiores como exteriores, el diseño eléctrico para la iluminación es de suma importancia. Este diseño no solo se basa en la distribución de conductos eléctricos, sino también en el alcance arquitectónico que se pueda tener en cuenta al estudiar las necesidades de iluminación en el proyecto.

Es importante considerar el diseño lumínico adecuado, ya que en el territorio peruano existen regulaciones normativas que establecen niveles mínimos de iluminación para diferentes tipos de edificaciones y usos de los espacios interiores. Cumplir con estas normas es fundamental para garantizar la seguridad y comodidad de los usuarios del espacio.

Sin embargo, es importante mencionar que Revit no cuenta con herramientas por defecto que permitan realizar un cálculo y análisis detallado de la iluminación. Además, el entorno de Revit tampoco proporciona un alto potencial para el *Code Checking* en el ámbito de la iluminación. Por lo tanto, existe la necesidad de implementar la interoperatividad con normas especializadas y utilizar herramientas externas para realizar el cálculo lumínico y asegurar el cumplimiento de las regulaciones vigentes.

#### 5.3.5.1 Base del Algoritmo

El objetivo programable para el desarrollo de herramientas dedicadas al diseño y cálculo de instalaciones eléctricas de iluminación es lograr un emplazamiento automático de luminarias que cumpla con los niveles mínimos de iluminación requeridos. En el entorno de Revit, el modelado de puntos de iluminación puede ser complejo, ya que implica manejar diferentes tipos de interacciones y restricciones dependiendo del tipo de luminaria y su relación con los elementos arquitectónicos, como techos y paredes.

Además, la manipulación masiva de elementos dentro de Revit, como la distribución de numerosas luminarias, puede ser limitada utilizando solo funciones de matrices o arreglos lineales o radiales. Por lo tanto, la herramienta a programar debe enfocarse en permitir la colocación automática de una gran cantidad de luminarias teniendo en cuenta el diseño de iluminación deseado.

En este contexto, la herramienta a desarrollar debe interoperar con software especializado en diseño lumínico, como *Dialux Evo*, que ofrece funcionalidades avanzadas para calcular y visualizar la iluminación en espacios arquitectónicos. La ventaja de utilizar *Dialux Evo* radica en su gratuidad, lo que permite desarrollar la programación sin incurrir en gastos de licencias u operación que podrían presentarse con otros softwares de diseño lumínico.

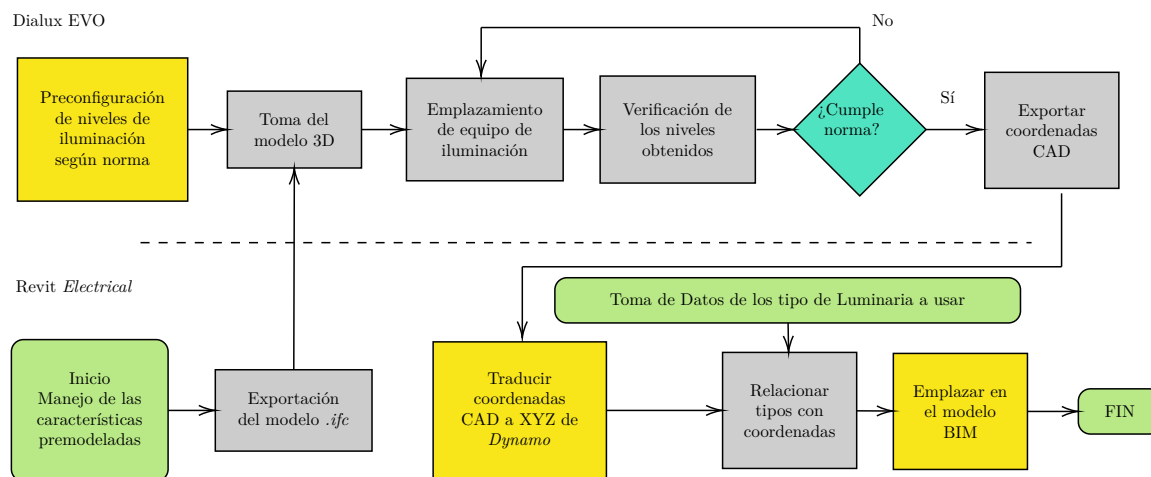
Sin embargo, es importante tener en cuenta que *Dialux Evo* solo tiene la capacidad de exportar información a formatos CAD. Por lo tanto, la base del algoritmo a desarrollar consiste en

adaptar el formato CAD proporcionado por *Dialux Evo* para que sea compatible con Revit. Este proceso de intermediación permitirá traer los resultados geométricos e informativos de *Dialux Evo* hacia Revit, asegurando una integración eficiente de los datos de iluminación en el proyecto eléctrico en Revit.

La Figura 5.12 muestra una división en el flujo que expresa la interoperatividad que se pretende implementar entre *Dialux EVO* y Revit. Aunque actualmente no existe una compatibilidad directa entre *Dialux EVO* y Dynamo u otra interfaz de programación paramétrica-visual, se plantea realizar la operación dentro de *Dialux EVO* de forma manual, desde la preconfiguración hasta la etapa de exportación de coordenadas.

**Figura 5.12**

*Diagrama de Flujo para el script “Iluminación Óptima”*



*Fuente:* Elaboración propia.

Para automatizar el proceso desde Dynamo, se puede interactuar con el software *AutoCAD 2022*, que tiene la capacidad de trabajar con coordenadas *XYZ*. A través de programación tradicional en el entorno de *AutoCAD*, es posible tomar las coordenadas desde el archivo de texto generado por *Dialux EVO*.

En Dynamo, se puede aprovechar la lectura de archivos de texto y el desenlazado de cadenas de texto para procesar la información de coordenadas y transformarla en valores numéricos. Con esta información, se puede realizar la traducción geométrica de los puntos de luz en el script.

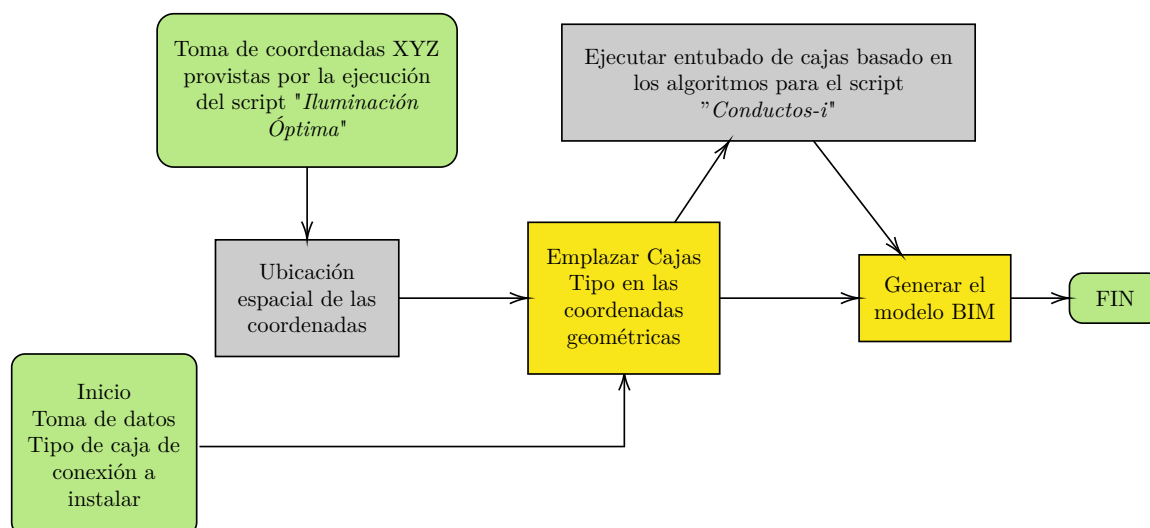
En casos reales, cada punto de luz en el diseño tiene especificaciones técnicas de instalación y montaje que deben adaptarse al alojamiento de empalmes y conexiones en cajas destinadas al uso eléctrico. Implementar esta característica adicional en el programa requeriría utilizar los nodos de manejo geométrico e instalación de ejemplares que ya posee Dynamo. Además, el entubado entre cajas también requeriría adaptar los algoritmos 5.1, 5.2, 5.3 y 5.4 a las nuevas necesidades.

Dentro del contexto de sistemas de iluminación modelados en Revit, al igual que en el entubado de sistemas de alimentación, se considera la necesidad de conectar las cajas que alojan empalmes o conexiones y buscar el entubado adecuado para dichos elementos. Esta tarea puede volverse compleja debido a la naturaleza del software BIM y la requerida creación de vistas auxiliares para entubados con características verticales y horizontales. En el caso específico del sistema de alumbrado, se enfrenta a situaciones similares cuando se necesitan realizar entubados en techos empotrados en elementos estructurales, lo que implica el uso de métodos de entubado tanto verticales como horizontales, además de emplear las curvas correspondientes.

Si el caso de uso para el sistema de iluminación es similar al primer script desarrollado para el entubado de sistemas de alimentación, como se muestra en el diagrama de flujo de la Figura 5.13, simplemente se requeriría adaptar los algoritmos previamente planteados para esta nueva situación.

**Figura 5.13**

*Diagrama de Flujo para el instalaciones de cajas de conexiones y entubado respectivo*

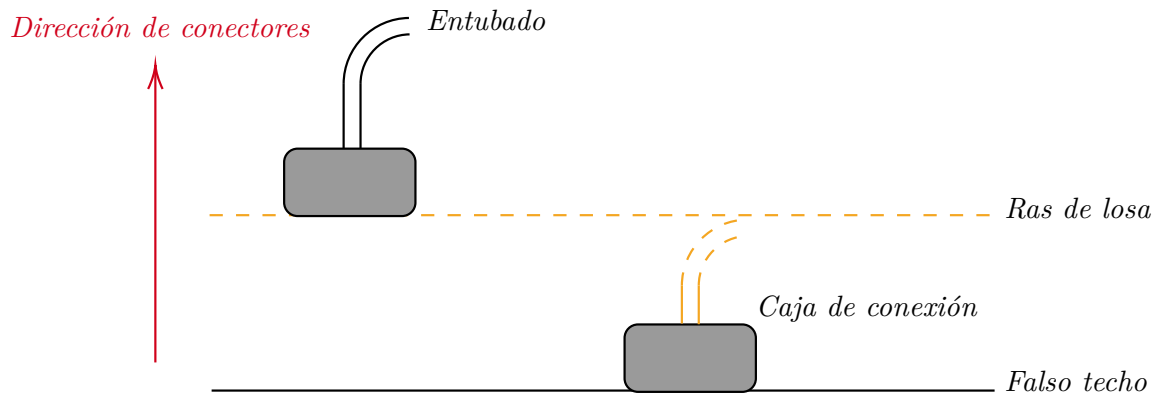


*Fuente:* Elaboración propia.

Dentro de la Figura 5.14 se pueden observar dos situaciones en las cuales se requieren entubados que involucren elementos tanto verticales como horizontales. En el caso de los entubados verticales, los conectores de Revit tendrán una dirección vectorial positiva en el eje  $Z$  ( $\vec{Z}_{Conectores} = +1$ ). Para abordar esta funcionalidad en Dynamo, se plantean los algoritmos 5.5 y 5.6.

Figura 5.14

Diagrama referencial para entubado de cajas de conexión para circuitos de iluminación



Fuente: Elaboración propia.

---

**Algoritmo 5.5** Cálculo del conector más próximo y generación de líneas para sistemas de iluminación

**Require:** Niveles (Lista), Puntos Conectores Superiores (Lista)

```

1: Lev ← Niveles
2: Sup ← Conectores Superiores por elemento
3: TamEle ← Tamaño de lista Lev
4:  $X_{sup}, Y_{sup}, Z_{sup}$  ← Coordenadas  $(X, Y, Z)$  de Sup
5: LIN ← lista de ceros de tamaño de TamLev
6: USED ← lista de ceros de tamaño de TamLev
7:  $X_o, Y_o, Z_o$  ← 0
8:  $X_{ff}, Y_{ff}, Z_{ff}$  ← 0
9: for  $i = 0 \dots (TamEle - 1)$  do
10:   LIN[i] ← Lin2_I(i, i + 1)
11: end for
    return LIN

```

Fuente: Elaboración propia.

---

---

**Algoritmo 5.6** Función *Lin2\_I*, parte del Algoritmo 5.5

---

**Require:** listas:  $Sup, X_{sup}, Y_{sup}, Z_{sup}, USED$

```

1: function LIN2_I( $a, b$ )
2:    $Min \leftarrow 10000$ 
3:    $M \leftarrow$  Vector inicializador vacío
4:    $P_i \leftarrow Sup[a]$ 
5:    $h \leftarrow$  Tamaño de la lista  $Sup[a]$ 
6:    $X_o, Y_o, Z_o \leftarrow X_{sup}[a], Y_{sup}[a], Z_{sup}[a]$ 
7:    $P_f \leftarrow Sup[b]$ 
8:    $w \leftarrow$  Tamaño de la lista  $Sup[b]$ 
9:    $X_{ff}, Y_{ff}, Z_{ff} \leftarrow X_{sup}[b], Y_{sup}[b], Z_{sup}[b]$ 
10:  for  $m = 1 \dots$  Tamaño de lista  $P_i$  do
11:    if  $USED[a] \neq 0$  then
12:      if  $USED[a] \cong P_i[m]$  es Verdadero then
13:        Extender lista  $M$  con  $P_i[m]$ 
14:        Quitar  $X_o[m], Y_o[m], Z_o[m]$ 
15:      end if
16:    end if
17:  end for
18:  for  $m = 1 \dots$  Tamaño de lista  $M$  do
19:    Quitar  $M[m]$  de lista  $P_i$ 
20:  end for
21:   $line \leftarrow 0$ 
22:  for  $i = 1 \dots$  Tamaño de lista  $P_i$  do
23:    for  $j = 1 \dots$  Tamaño de lista  $P_f$  do
24:       $X_i, Y_i, Z_i \leftarrow X_o[i], Y_o[i], Z_o[i]$ 
25:       $X_f, Y_f, Z_f \leftarrow X_{ff}[j], Y_{ff}[j], Z_{ff}[j]$ 
26:       $DIS \leftarrow \sqrt{(X_f - X_o)^2 + (Y_f - Y_o)^2 + (Z_f - Z_o)^2}$ 
27:      if  $DIS < Min$  then
28:         $line \leftarrow$  Generar línea entre  $P_i[i]$  y  $P_f[j]$ 
29:         $USED[a, i] \leftarrow P_i[i]$ 
30:         $Min \leftarrow DIS$ 
31:      end if
32:    end for
33:  end for
34:  return  $line$ 
35: end function

```

*Fuente:* Elaboración propia.

---

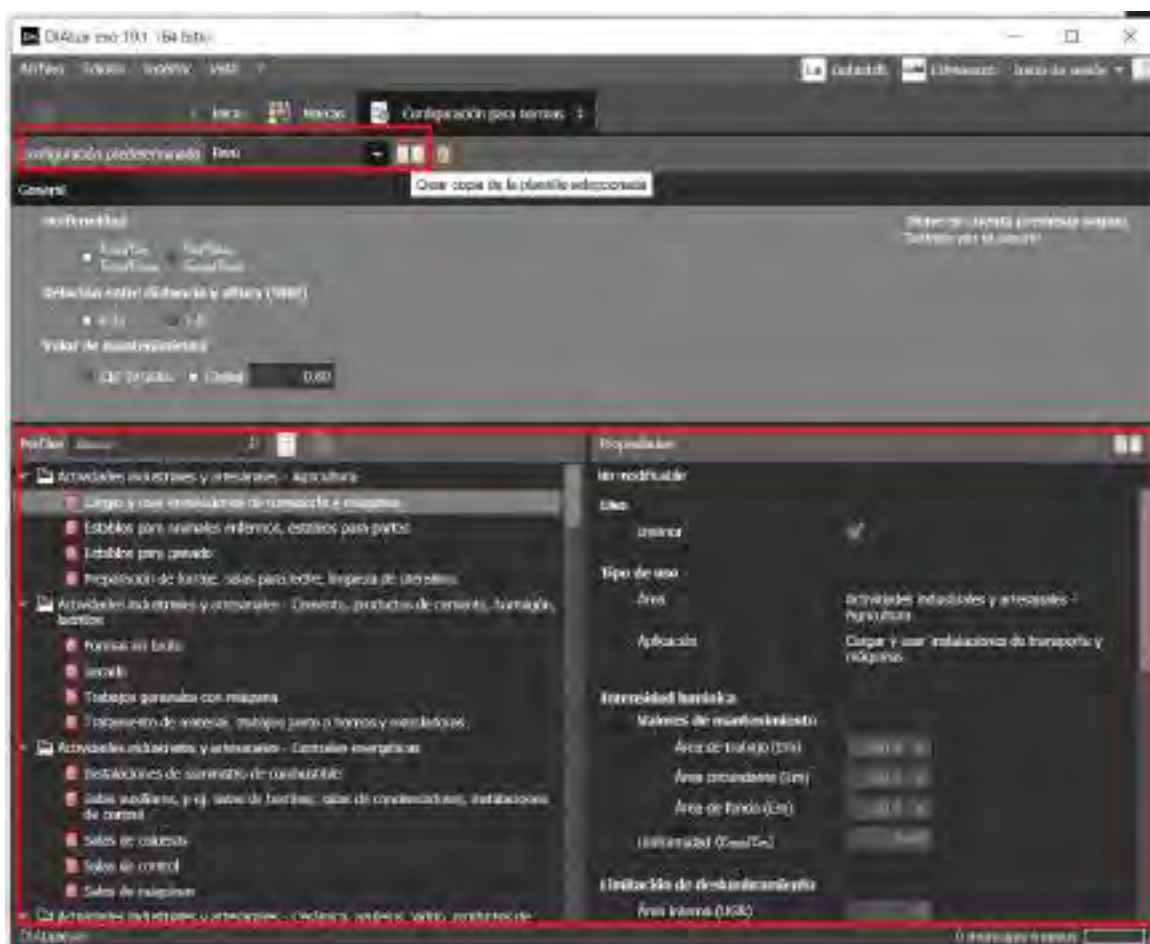
### 5.3.5.2 Desarrollo del Script “Iluminación Óptima”

Para el desarrollo del script “Iluminación Óptima”, se abordará en primer lugar la operación con Dialux EVO y su relación con la normativa peruana. Se comenzará estableciendo los niveles mínimos de iluminación, que serán los estándares indicados por la norma peruana Reglamento Nacional de Edificaciones (RNE), en su especialidad de Eléctricas y Mecánicas (Ministerio de Vivienda, Construcción y Saneamiento, 2006).

La norma *EM.010*, que forma parte del RNE y fue actualizada en 2019 mediante la Resolución Ministerial N° 083-2019-VIVIENDA, incluye en sus anexos una tabla con los niveles mínimos de iluminancia, índice de deslumbramiento, y observaciones sobre el tipo de ambiente. Estos datos servirán de base normativa para la configuración de Dialux EVO y la generación de nuevos perfiles normativos que cumplan con los estándares peruanos (Ver Figura 5.15).

**Figura 5.15**

*Configuración de nuevo perfil normativo en Dialux EVO*

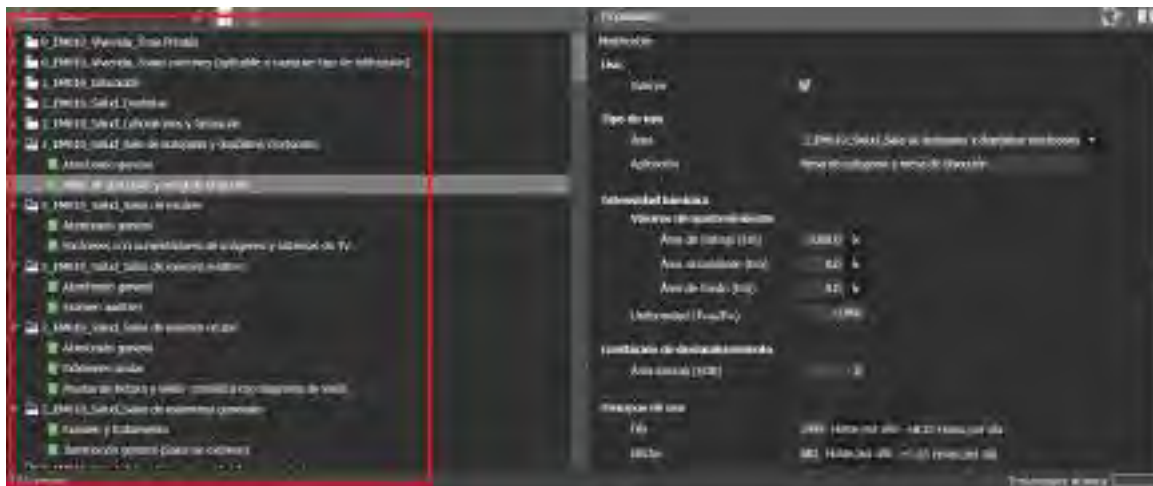


Fuente: Captura de software propia.

Aunque Dialux EVO se basa en normas europeas, que sustentan sus estándares, es necesario especificar los requerimientos de la normativa peruana en muchos casos para asegurar el cumplimiento de los estándares locales (Ver Figura 5.16).

**Figura 5.16**

*Generación de nuevos perfiles en base a la norma EM.010 del RNE*



*Fuente:* Captura de software propia.

A continuación, en el proceso algorítmico, se destaca la interacción entre Revit y Dialux EVO, que es posible gracias al formato *.ifc*, permitiendo migrar la información BIM hacia el software objetivo. Sin embargo, es importante mencionar que al momento de redactar esta tesis y con la versión 10.1 del software de DIAL GmbH, la funcionalidad de manejo de formatos *.ifc* está aún en condición “beta” (Ver Figura 5.17). Debido a esto, en modelos BIM muy complejos, existe la posibilidad de enfrentar limitaciones y pérdida de información en ciertos casos.

Dentro de Dialux EVO, una vez realizado el diseño lumínico requerido y obtenidos los resultados más satisfactorios, el software ofrece la posibilidad de exportar los resultados en un formato específico, que es el *.dwg* correspondiente a dibujo vectorial CAD.

La interfaz de exportación es vital para asegurar la correcta distinción de la capa correspondiente al artefacto de iluminación. En la Figura 5.18 se muestra una configuración básica recomendada para facilitar el trabajo de toma de coordenadas en AutoCAD.

Dentro del nuevo software, se observa que las capas tienen una codificación bastante básica para mapear las luminarias respectivas según su ubicación precisa dentro de la edificación previamente modelada en Revit y calculada lumínicamente en Dialux EVO. En la interfaz de este software, estos elementos se denominan como “bloques”, y cada uno de ellos tiene una propiedad de ubicación única en el espacio tridimensional (Ver Figura 5.19).

Es importante destacar que se están utilizando las mismas coordenadas en cada proceso de interoperatividad entre Revit, Dialux EVO y AutoCAD. Esto permite mantener la consistencia

**Figura 5.17**

*Aviso sobre los modelos .ifc dentro de Dialux EVO*



*Fuente:* Captura de software propia.

y precisión en el posicionamiento de las luminarias en el modelo, facilitando la integración de la información BIM y los resultados del cálculo lumínico en el software de diseño eléctrico.

El nuevo proceso de programación busca optimizar la obtención de coordenadas de cada luminaria desde AutoCAD, ya que el proceso manual de exportar coordenadas por cada bloque es complicado y tedioso. Para simplificar esta tarea, se recurre a la programación tradicional y se utiliza una herramienta desarrollada por Tharwat (2010) llamada “Export Block Coordinates” (Consultar el código completo en el Anexo B - Código A.9).

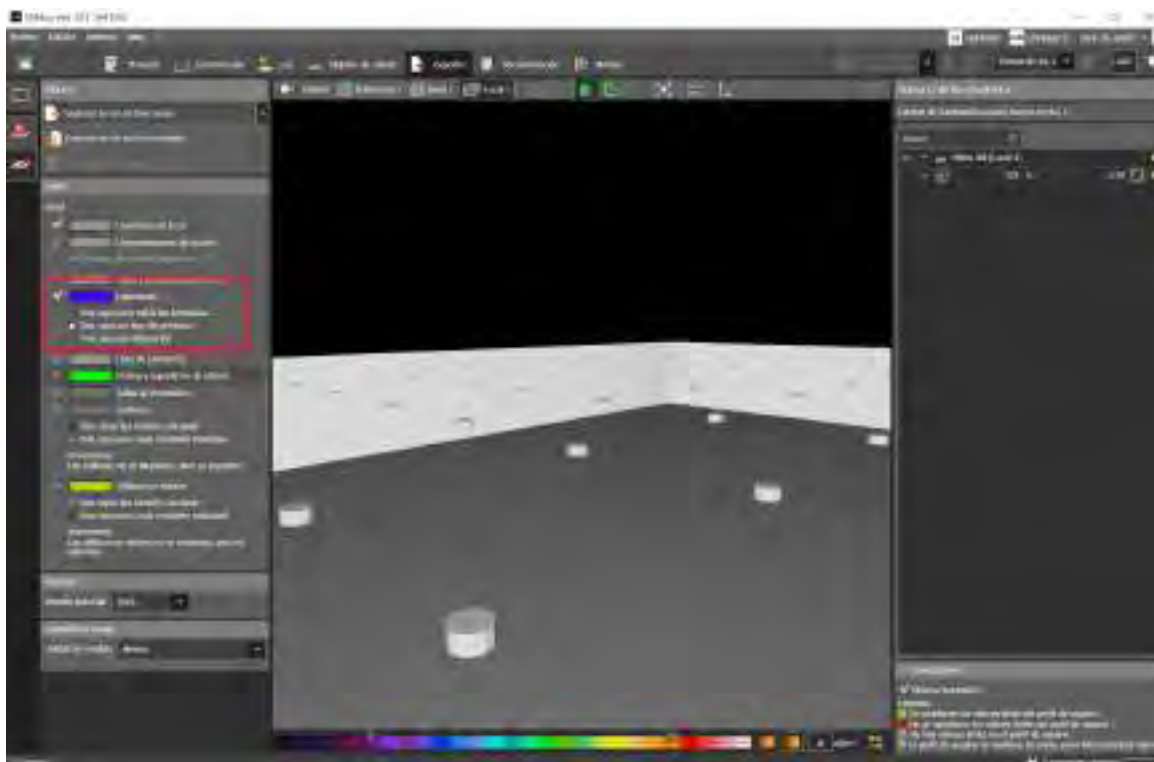
Esta herramienta permite la exportación de las coordenadas *XYZ* de cada elemento seleccionado en AutoCAD hacia un archivo de texto básico en formato *.txt*. El archivo resultante se ubica en la misma carpeta que el archivo CAD y contiene la información de las coordenadas de los bloques seleccionados (Ver Figura 5.20). Con esta funcionalidad gestionable de archivos, se agiliza y automatiza el proceso de obtener las coordenadas de las luminarias para su posterior uso en el desarrollo del script en Dynamo.

A partir de este archivo de coordenadas en formato de texto, es posible extraer una gran cantidad de información mediante el manejo programable de cadenas de texto. Si es necesario convertir texto en valores numéricos, se puede entrar al campo de tareas que se pueden desarrollar en la interfaz programable de Dynamo. Esto es precisamente lo que se continúa en el diagrama de flujo propuesto para esta sección, ya que con las coordenadas en formato de texto, es posible traducirlas a las coordenadas que maneja Revit para el proyecto en el cual se está trabajando.



**Figura 5.18**

*Configuración básica para la exportación de datos de luminarias a formato .dwg*



*Fuente:* Captura de software propia.

Al igual que en el caso programable anterior, para esta herramienta se recurrirá al uso de nodos desarrollados por terceros, los cuales son los siguientes:

- Paquetes Adicionales:
  1. *Data-Shapes* por Elayoubi Mostafa
  2. *Crumple* por Gavin Crump<sup>4</sup>
  3. *Orchid* por erfajo<sup>5</sup>
  4. *Springs Nodes* por dimitar.ven
  5. *KYSUDO* por kstheanhdo<sup>6</sup>

Dentro del proceso planteado inicialmente para la operación dentro de Dynamo se encuentra la tarea de traducir las coordenadas obtenidas por la ejecución del comando creado por Tharwat (2010) dentro de AutoCAD. Para este fin, Dynamo posee nodos por defecto que se pueden usar para extraer las cadenas de texto precisas. La Figura 5.21 muestra el proceso de desintegración de cadenas de texto correspondientes al proyecto.

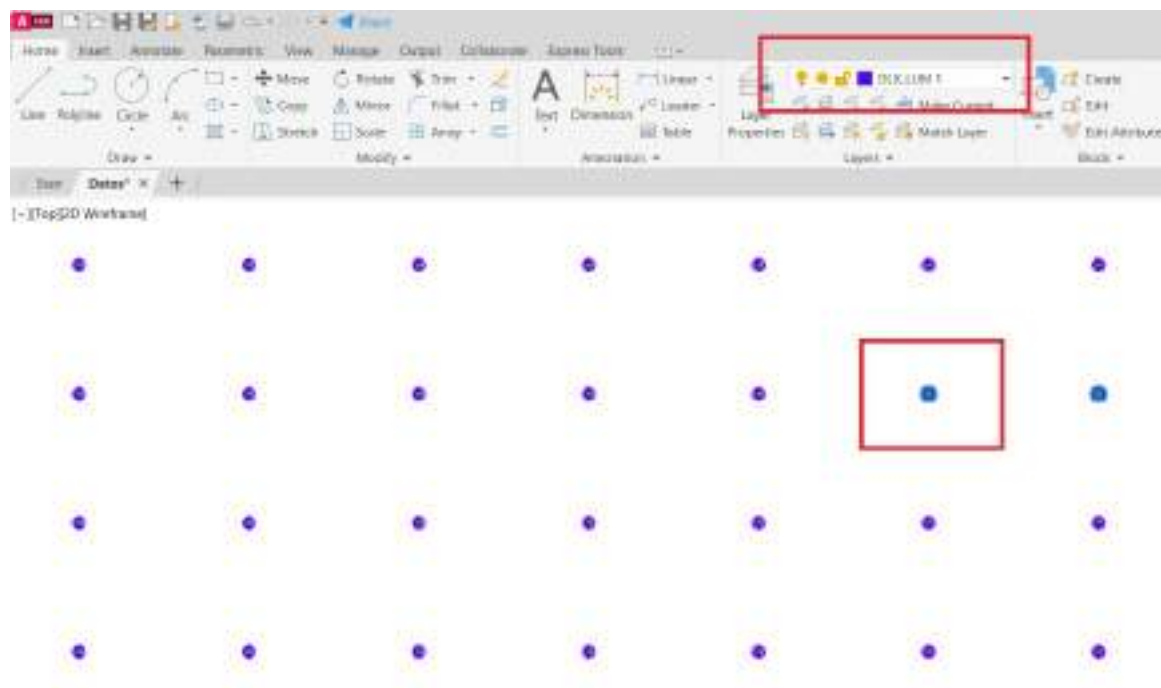
<sup>4</sup> Sitio Web del autor: <https://www.binguru.com.au/>

<sup>5</sup> Repositorio del autor: <https://github.com/erfajo/>

<sup>6</sup> Sitio Web del autor: <https://www.kysudo.com>

Figura 5.19

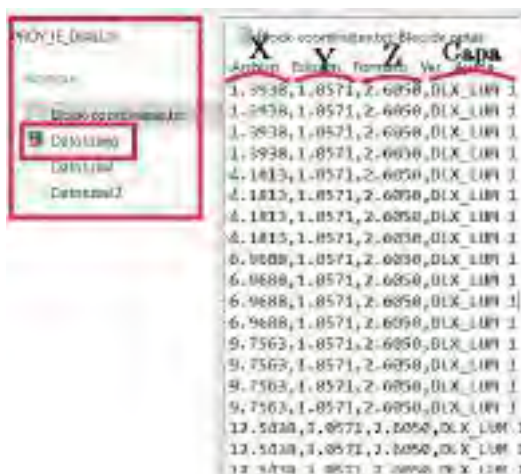
Mapeado de Luminarias por capas en AutoCAD



Fuente: Captura de software propia.

Figura 5.20

Generación de archivo de texto con coordenadas de luminarias



Fuente: Captura de software propia.

Figura 5.21

Extracción de coordenadas XYZ de luminarias hacia listas Dynamo



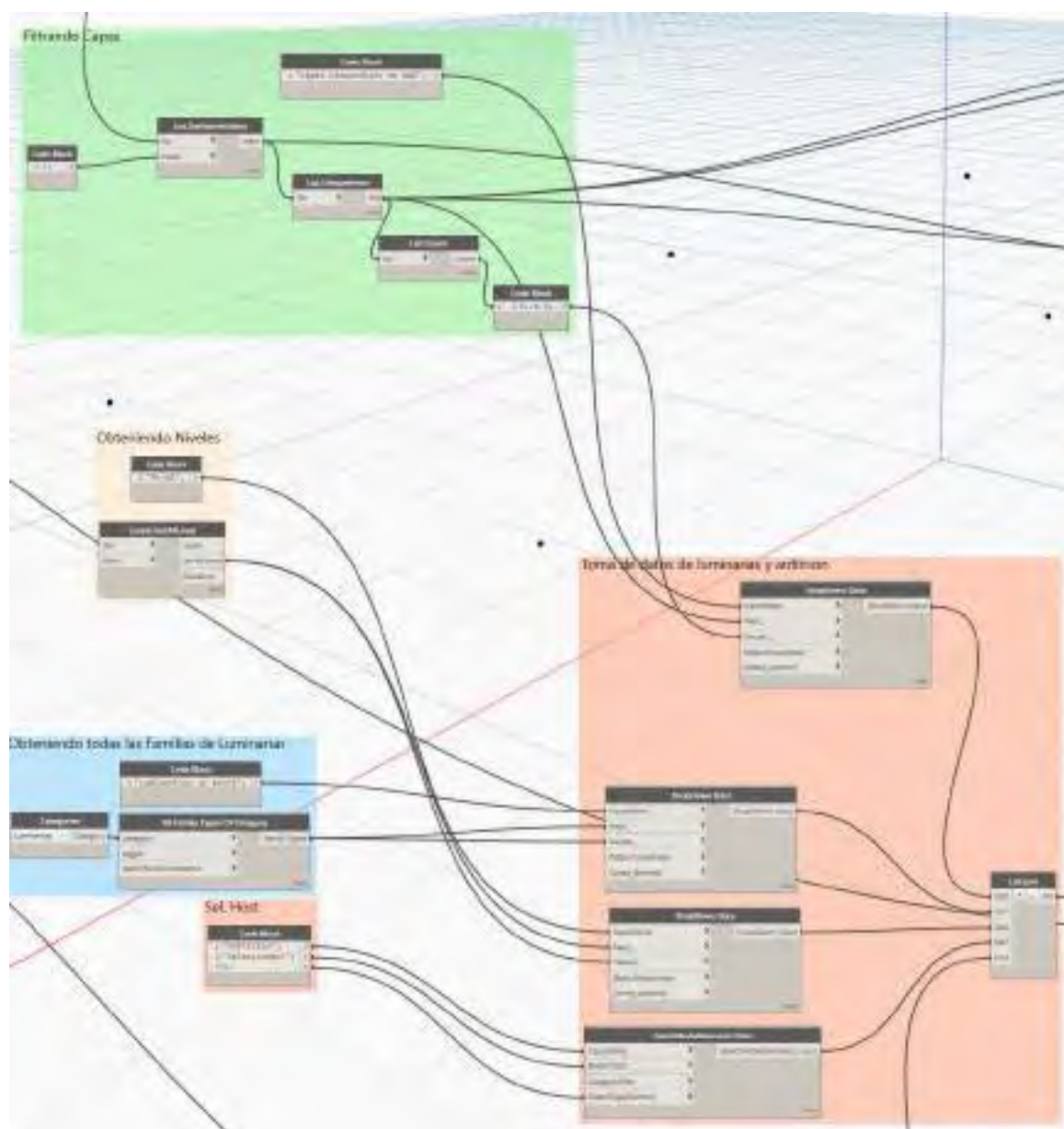
Fuente: Implementación propia.

Con la diferenciación de coordenadas  $X$ ,  $Y$  y  $Z$ , así como la identificación de las capas obtenidas del archivo *.dwg* producto de la exportación de resultados desde Dialux EVO, dentro de Dynamo se puede asignar una lista de tipos de luminarias para reemplazar cada capa. Es importante tener en cuenta que los formatos utilizados en Revit y Dialux EVO son distintos, por lo que no se pueden reutilizar directamente. Sin embargo, dado que existen bibliotecas BIM públicas, es posible tomar los archivos necesarios sin recurrir a repositorios adicionales al del fabricante original.

Considerando todo lo expuesto anteriormente, en la Figura 5.22 se presenta una segunda interfaz de selección de datos que se pueden obtener en el mismo modelo de Revit.

**Figura 5.22**

*Toma de datos para las luminarias y su respectiva capa generada por Dialux EVO*



*Fuente:* Implementación propia.

Además de los parámetros BIM asociados a cada luminaria, es necesario asignar un nivel de referencia para el elemento, con el propósito de una futura transición hacia dimensiones superiores de planificación, cantidades y costos.

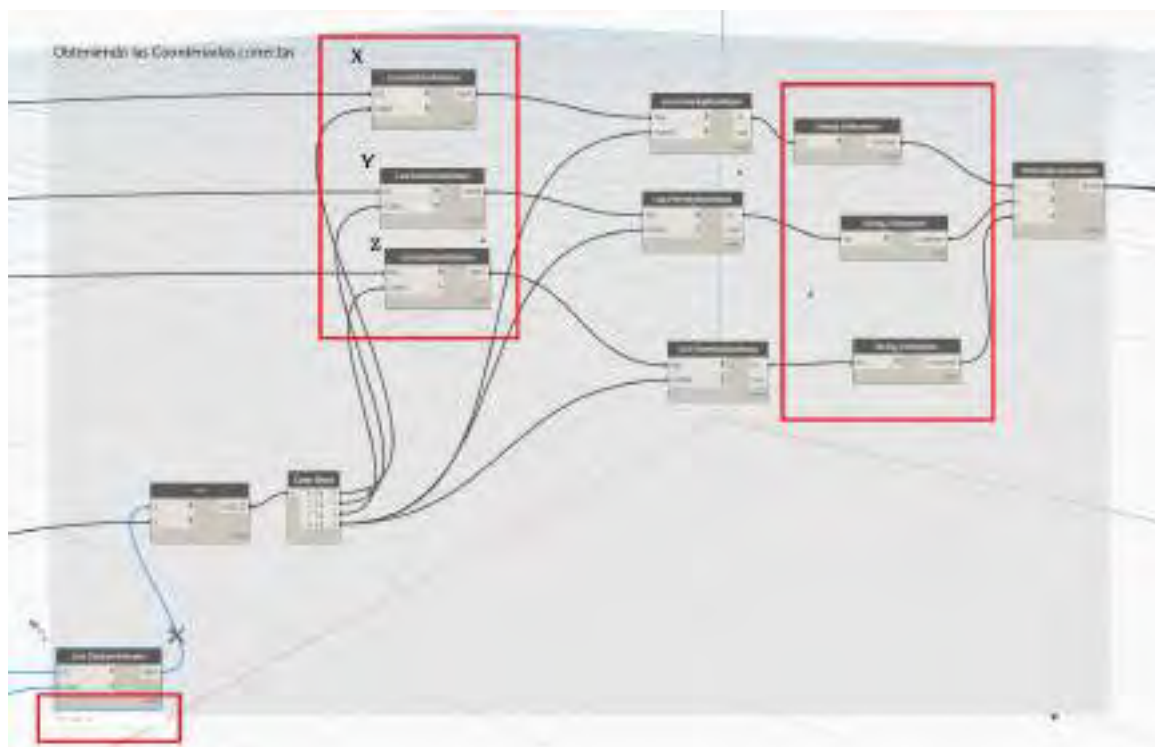
En relación con el método adecuado de emplazamiento de las luminarias (por cara, por anfitrión o por plano de trabajo), se ha decidido optar por la selección de un primer anfitrión. En caso de que este método falle, se pueden aplicar nodos adicionales para prevenir el mal funcionamiento, pero en principio, se dará prioridad a la primera toma de datos que se dirige hacia el anfitrión.

Una vez obtenidos los datos básicos, el primer paso es importar la información de la capa correspondiente. Dado que en la etapa anterior se han filtrado y enlistado los datos según la capa y la luminaria, los filtros por máscara booleana permiten que se obtengan solo los datos que cumplen con el criterio especificado, es decir, los correspondientes a las capas correctas.

En la Figura 5.23, se puede observar que al aplicar el filtro con la capa, se obtienen las coordenadas  $X$ ,  $Y$  y  $Z$  una a una. Es importante tener en cuenta que, en sistemas programables, un número puede ser representado como una cadena de texto y, por lo tanto, no se pueden realizar operaciones geométricas directamente con cadenas de texto.

**Figura 5.23**

*Generación de puntos geométricos para luminarias por capa de Dialux EVO*



*Fuente:* Implementación propia.

Por esta razón, es necesario convertir cualquier cadena de texto a un dato de tipo numérico. Solo cuando los datos estén en forma numérica, será posible generar puntos geométricos a través de las coordenadas *XYZ* obtenidas.

En cuanto a los métodos de emplazamiento, como se explicó anteriormente, es importante prever distintos enfoques en caso de que el emplazamiento falle en un primer intento. Aunque los procesos iterativos o recursivos no son posibles directamente en Dynamo, los nodos condicionales funcionan de manera similar a la programación tradicional y sí son aplicables mediante los nodos por defecto de Dynamo.

Por lo tanto, se puede utilizar nodos condicionales para alternar entre distintos métodos de emplazamiento en función de las circunstancias. Si un método falla, se puede utilizar un nodo condicional para pasar al siguiente método y así sucesivamente hasta encontrar el método que funcione correctamente.

Para facilitar la identificación de errores y la toma de decisiones, se pueden utilizar mensajes de error que se activan en caso de que un método de emplazamiento no sea exitoso. La Figura 5.24 muestra un ejemplo de cómo se pueden implementar estos nodos condicionales y mensajes de error en el proceso de emplazamiento de luminarias.

Como parte del aprovechamiento de los puntos geométricos ya obtenidos, es necesario tener en cuenta que en el mismo lugar donde se emplazan las luminarias también deben instalarse las cajas que alojan las conexiones y empalmes eléctricos. En el caso de que estas cajas se instalen empotradas en elementos de techo o losas estructurales, el método de emplazamiento es similar a la forma en que se realizó con las luminarias.

Sin embargo, también existen ocasiones en las que, dependiendo de las propiedades del elemento a emplazar, pueda fallar la instalación por anfitrión o por cara. En tales casos, la operación programable toma decisiones a través del uso de nodos condicionales, permitiendo alternar entre distintos métodos de emplazamiento para lograr una instalación exitosa. La Figura 5.25 muestra un ejemplo de cómo se pueden utilizar estos nodos condicionales para abordar diferentes situaciones de emplazamiento de cajas eléctricas.

Figura 5.24

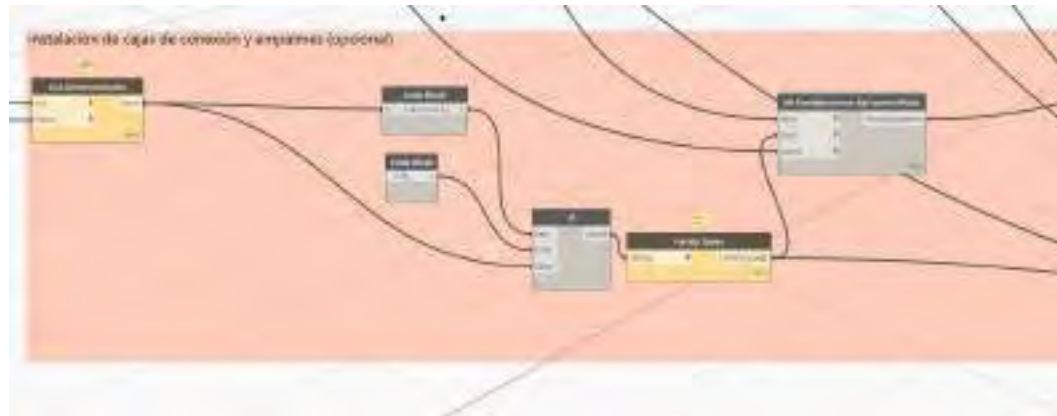
Métodos de emplazamiento de luminarias en Dynamo



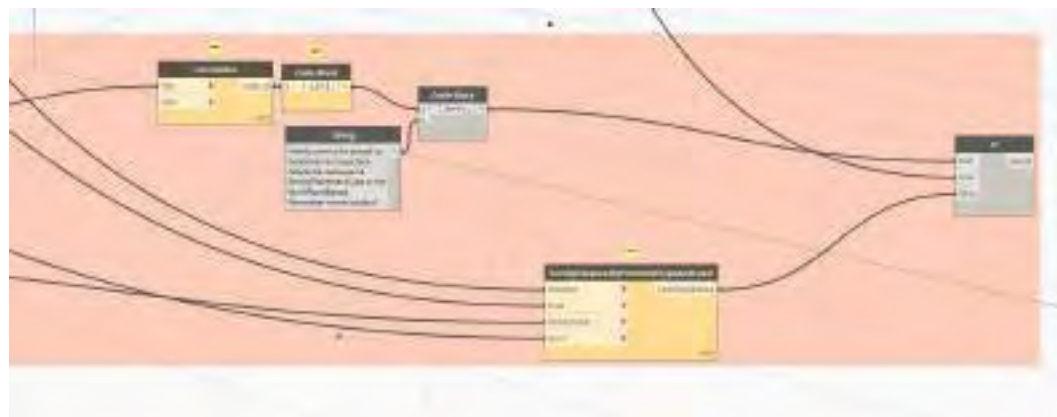
Fuente: Implementación propia.

**Figura 5.25**

*Instalación de cajas de conexión y empalmes para luminarias emplazadas*



(a) *Primera parte del script*



(b) *Segunda parte del script*

*Fuente:* Implementación propia.



### 5.3.5.3 Desarrollo del Script “Conductos I”

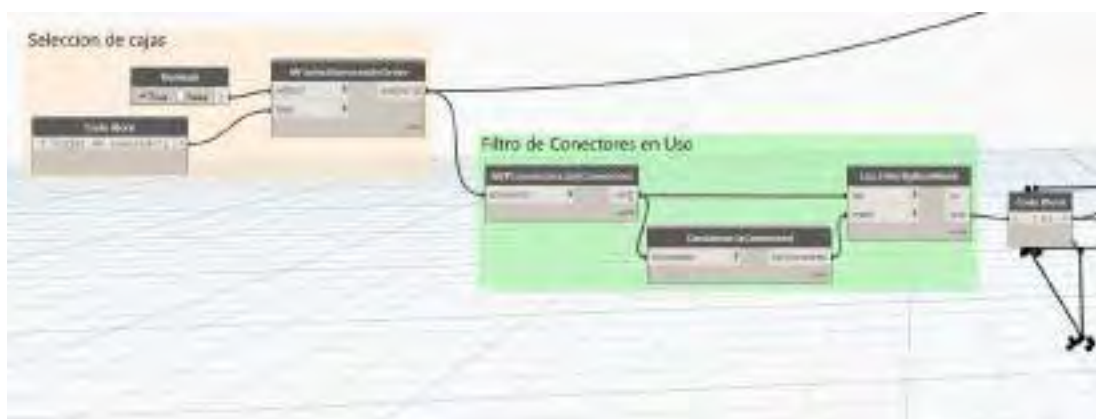
En el desarrollo del script “Conductos I”, se debe tener en cuenta que se están adaptando los algoritmos 5.1 y 5.2. Por lo tanto, la entrada de datos hacia el sistema de programación de Dynamo debe ser similar, respetando el orden en el que se tiene la selección de cajas de conexiones.

El mapeado de conectores se realiza de manera similar al script “Conductos-i”, pero en este caso se reduce a considerar solo los conectores superiores, ya que será en estos donde se llevará a cabo el entubado en esquemas de falsos techos o elementos estructurales.

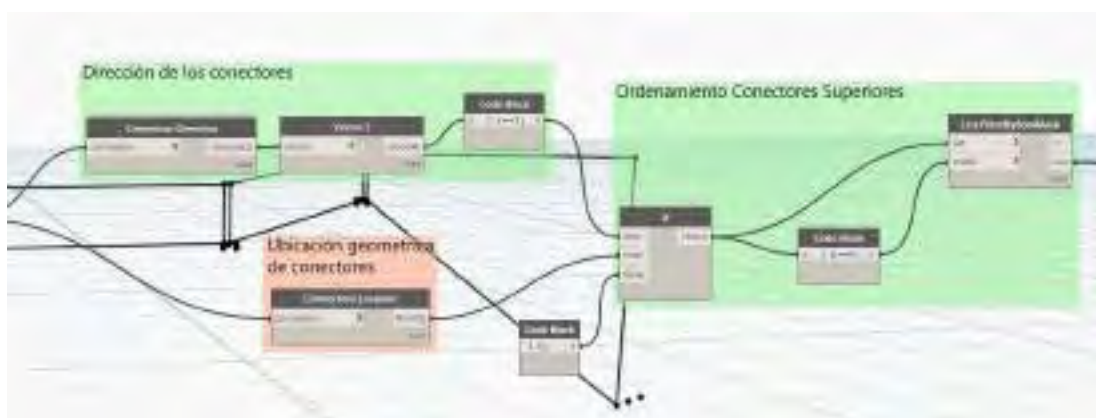
El proceso de manejo de listas para un uso masivo de datos y la conversión de ubicaciones a puntos geométricos continúa en este script (Ver Figura 5.26). También se reducen los nodos de extracción de coordenadas  $XYZ$  debido a que en esta ocasión solo se considera el entubado en una sola dirección vertical.

**Figura 5.26**

*Entrada de datos y filtro de conectores para el script Conductos I*



(a) Primera parte del script



(b) Segunda parte del script

*Fuente:* Implementación propia.

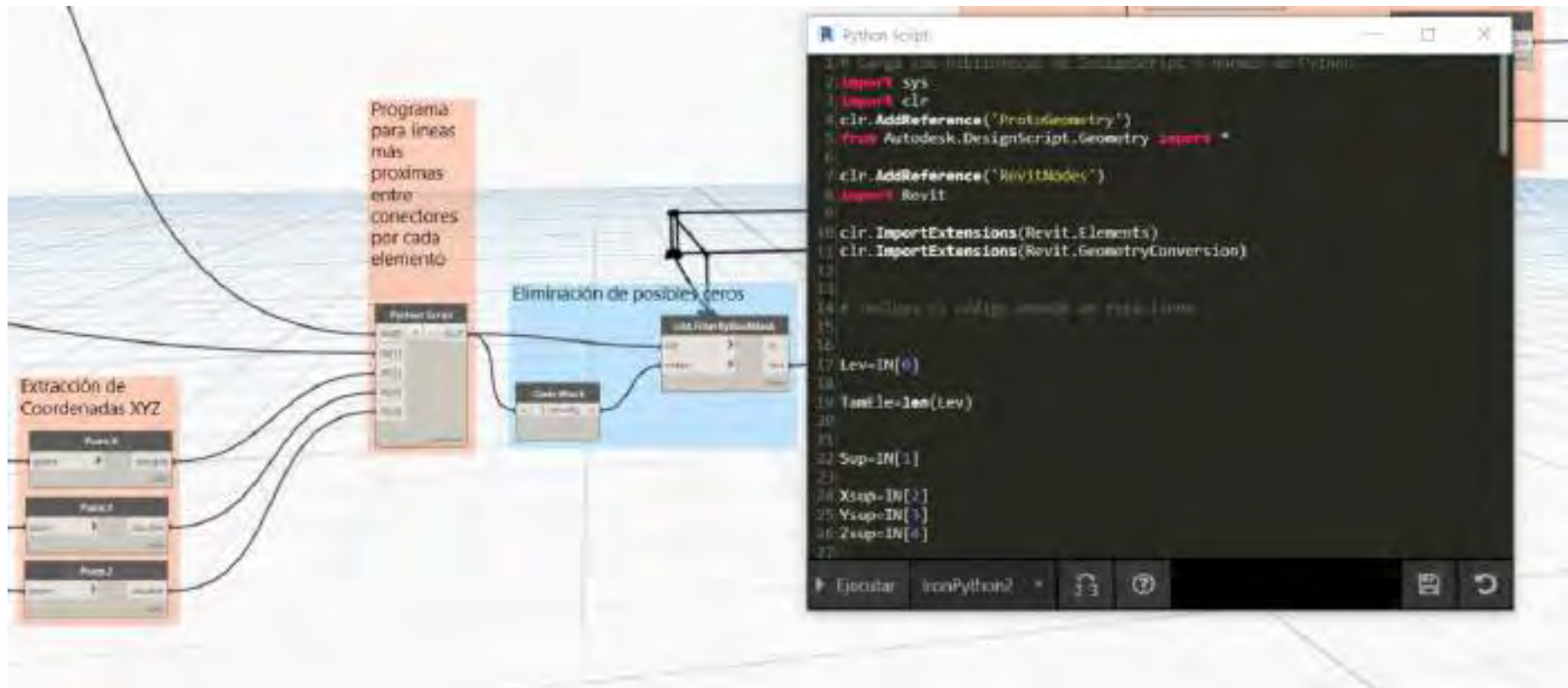
Luego, se implementan los algoritmos 5.5 y 5.6 a través de la programación tradicional en lenguaje Python (Ver Figura 5.27). El código fuente completo de esta implementación se encuentra en el Anexo A - Código A.3.

En la Figura 5.28, con los puntos de conectores más próximos a cada caja calculados como entrada, se realiza la traslación en el eje  $Z$  para generar líneas verticales, lo que no requiere la aplicación de otro nodo de programación de Python. A través de nodos, se puede obtener un resultado rápido debido a la reducción de la carga computacional del objetivo a programar.

Finalmente, el trabajo restante implica la transición de líneas geométricas hacia elementos reales, que en este caso son los ejemplares de tubos a usarse, así como las curvas necesarias para el entubado correcto. Al igual que en el script “Conductos-i”, el método de operación es similar, ya que se recurre a la misma interfaz de selección del tipo de tubo y el tamaño de su diámetro (Ver Figura 5.29).

Figura 5.27

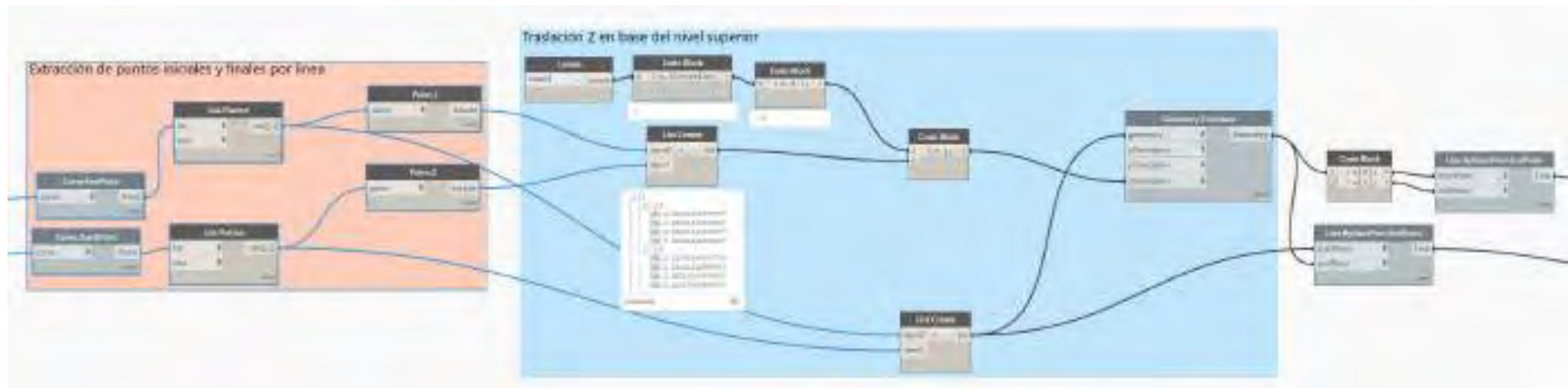
Implementación de los algoritmos 5.5 y 5.6 en Dynamo



Fuente: Implementación propia.

**Figura 5.28**

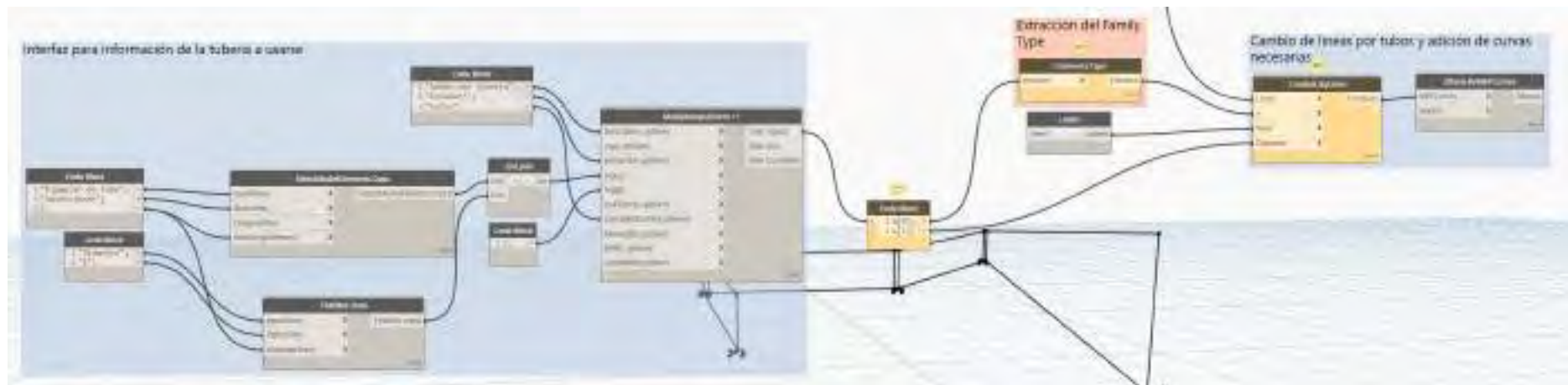
*Traslación Z para la generación de líneas sin uso de programación tradicional*



*Fuente:* Implementación propia.

**Figura 5.29**

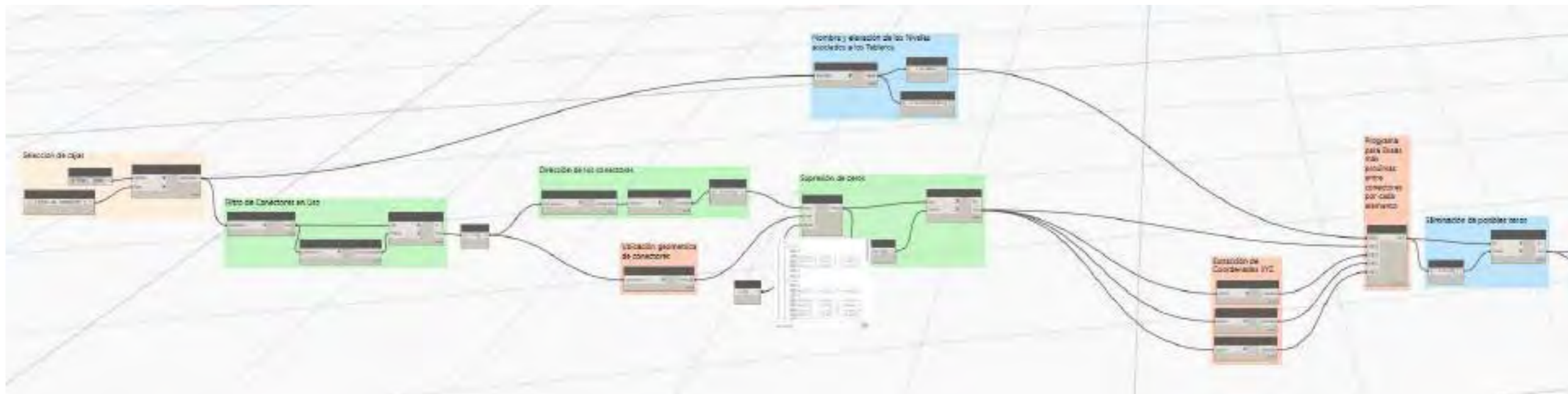
*Generación de tubos a partir de líneas para sistemas de alumbrado*



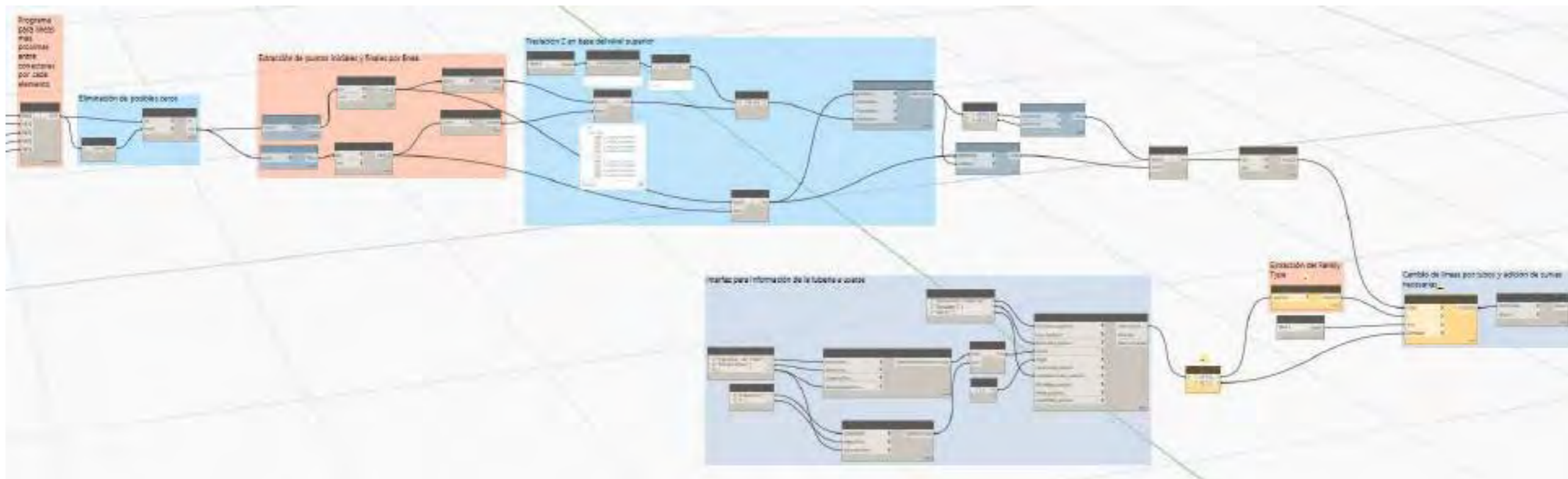
*Fuente:* Implementación propia.

Figura 5.30

Vista completa del script Conductos-I



(a) Primera parte del script



(b) Segunda parte del script

Fuente: Implementación propia.

### 5.3.6 Tomacorrientes

El diseño de instalaciones eléctricas interiores incluye el emplazamiento de tomacorrientes, que son puntos de acceso para obtener energía eléctrica para equipos que requieran alimentación, ya sea monofásica o trifásica.

En el proceso de diseño, se toma en cuenta el estudio de necesidades y las características usuales de uso de los tomacorrientes, basándose en el mobiliario previsto por el arquitecto proyectista.

En cuanto a los aspectos programables, se prioriza la capacidad de permitir el emplazamiento masivo de unidades de tomacorrientes en el software BIM. Además, es importante diferenciar entre la placa de tomacorrientes y la caja de conexiones al llevar el diseño de estas instalaciones a la metodología BIM.

El término internacionalmente conocido como “outlet” corresponde precisamente a la placa de tomacorrientes, que contiene las características eléctricas como la corriente nominal y la tensión de operación. Por otro lado, la caja de conexiones es un elemento adicional para la proyección de instalaciones eléctricas interiores y también forma parte del proceso de emplazamiento.

Es esencial tener en cuenta tanto el equipo eléctrico (placa de tomacorrientes) como su caja de conexiones en el diseño de instalaciones eléctricas interiores. En consecuencia, también es importante considerar su respectivo entubado para asegurar una instalación correcta y segura.

#### 5.3.6.1 Base del Algoritmo

El desarrollo de los programas propuestos se enfocará en automatizar el emplazamiento de cajas de conexión basado en la ubicación previa de los tomacorrientes. Esto se debe a que, en tareas de diseño, los tomacorrientes no siguen una regla o normativa específica que limite su ubicación, a menos que se trate de ambientes particulares o especificaciones del arquitecto o propietario.

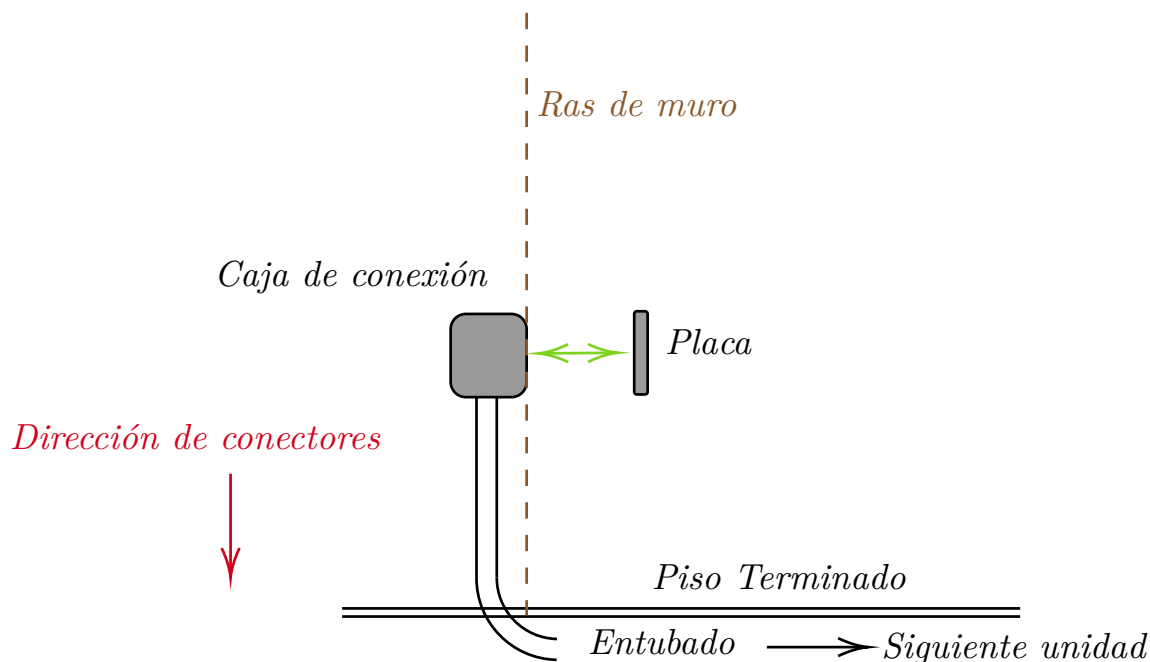
En el primer programa, se aprovechará el hecho de que los tomacorrientes ya están emplazados en el modelo para realizar el emplazamiento automatizado de las cajas de conexión. Esto permitirá una colocación eficiente y coherente de las cajas en relación con los tomacorrientes.

Por otro lado, en el segundo programa se utilizarán los algoritmos desarrollados previamente para sistemas de alimentación y sistemas de iluminación (algoritmos 5.1 y 5.2) para llevar a cabo el entubado por suelo de los tomacorrientes. Esta funcionalidad aprovechará las capacidades de los algoritmos existentes y los adaptará para el entubado de los tomacorrientes de manera eficiente y precisa.

En situaciones donde el usuario prefiera realizar el entubado por pared de forma directa, se obviará el caso programable. Esto se debe a que, en este escenario, no se requerirá el uso de vistas auxiliares en simultáneo dentro del software BIM, lo que simplificará el proceso.

**Figura 5.31**

Diagrama referencial para el emplazamiento de cajas de conexión y entubado de circuitos de tomacorrientes



Fuente: Elaboración propia.

La Figura 5.31 muestra un diagrama referencial que destaca la diferencia principal en el direccionamiento vectorial de los conectores de las cajas de conexión destinadas para los tomacorrientes en comparación con los sistemas de iluminación por techo. En este caso, los conectores de las cajas tienen una dirección vertical negativa ( $\vec{Z}_{Conectores} = -1$ ).

Dado que ya se adaptó con éxito la funcionalidad de los algoritmos 5.1 y 5.2 para los sistemas de iluminación por techo, se puede lograr el mismo resultado exitoso para los entubados por piso que involucran los tomacorrientes.

Aunque los algoritmos mencionados originalmente permitían prever el entubado en diferentes niveles, para mantener el orden y minimizar el esfuerzo computacional durante la ejecución de los scripts, se adaptará nuevamente la propuesta inicial (Ver algoritmos 5.7 y 5.8) para el caso actual. Esta adaptación permitirá realizar el entubado por piso de manera eficiente y precisa, tomando en cuenta el direccionamiento vertical negativo de los conectores de las cajas de conexión destinadas para los tomacorrientes.



---

**Algoritmo 5.7** Cálculo del conector más próximo y generación de líneas para sistemas de tomacorrientes

---

**Require:** Niveles (Lista), Puntos Conectores Inferiores (Lista)

```

1: Lev  $\leftarrow$  Niveles
2: Inf  $\leftarrow$  Conectores Inferiores por elemento
3: TamEle  $\leftarrow$  Tamaño de lista Lev
4:  $X_{inf}, Y_{inf}, Z_{inf} \leftarrow$  Coordenadas ( $X, Y, Z$ ) de Inf
5: LIN  $\leftarrow$  lista de ceros de tamaño de TamLev
6: USED  $\leftarrow$  lista de ceros de tamaño de TamLev
7: for  $i = 1 \dots TamLev$  do
8:   USED[ $i$ ]  $\leftarrow$  lista de ceros de tamaño de Inf
9: end for
10:  $X_o, Y_o, Z_o \leftarrow 0$ 
11:  $X_{ff}, Y_{ff}, Z_{ff} \leftarrow 0$ 
12: for  $i = 0 \dots (TamEle - 1)$  do
13:   LIN[ $i$ ]  $\leftarrow Lin2\_T(i, i + 1)$ 
14: end for
   return LIN

```

*Fuente:* Elaboración propia.

---

**Algoritmo 5.8** Función *Lin2\_T*, parte del Algoritmo 5.7

---

**Require:** listas:  $Inf, X_{inf}, Y_{inf}, Z_{inf}, USED$

```

1: function LIN2_T( $a, b$ )
2:    $Min \leftarrow 10000$ 
3:    $M \leftarrow$  Vector inicializador vacío
4:    $P_i \leftarrow Inf[a]$ 
5:    $h \leftarrow$  Tamaño de la lista  $Inf[a]$ 
6:    $X_o, Y_o, Z_o \leftarrow X_{inf}[a], Y_{inf}[a], Z_{inf}[a]$ 
7:    $P_f \leftarrow Sup[b]$ 
8:    $w \leftarrow$  Tamaño de la lista  $Inf[b]$ 
9:    $X_{ff}, Y_{ff}, Z_{ff} \leftarrow X_{inf}[b], Y_{inf}[b], Z_{inf}[b]$ 
10:  for  $m = 1 \dots$  Tamaño de lista  $P_i$  do
11:    if  $USED[a] \neq 0$  then
12:      if  $USED[a] \approx P_i[m]$  es Verdadero then
13:        Extender lista  $M$  con  $P_i[m]$ 
14:        Quitar  $X_o[m], Y_o[m], Z_o[m]$ 
15:      end if
16:    end if
17:  end for
18:  for  $m = 1 \dots$  Tamaño de lista  $M$  do
19:    Quitar  $M[m]$  de lista  $P_i$ 
20:  end for
21:   $line \leftarrow 0$ 
22:  for  $i = 1 \dots$  Tamaño de lista  $P_i$  do
23:    for  $j = 1 \dots$  Tamaño de lista  $P_f$  do
24:       $X_i, Y_i, Z_i \leftarrow X_o[i], Y_o[i], Z_o[i]$ 
25:       $X_f, Y_f, Z_f \leftarrow X_{ff}[j], Y_{ff}[j], Z_{ff}[j]$ 
26:       $DIS \leftarrow \sqrt{(X_f - X_o)^2 + (Y_f - Y_o)^2 + (Z_f - Z_o)^2}$ 
27:      if  $DIS < Min$  then
28:         $line \leftarrow$  Generar línea entre  $P_i[i]$  y  $P_f[j]$ 
29:         $USED[a, i] \leftarrow P_i[i]$ 
30:         $Min \leftarrow DIS$ 
31:      end if
32:    end for
33:  end for
34:  return  $line$ 
35: end function

```

*Fuente:* Elaboración propia.

---

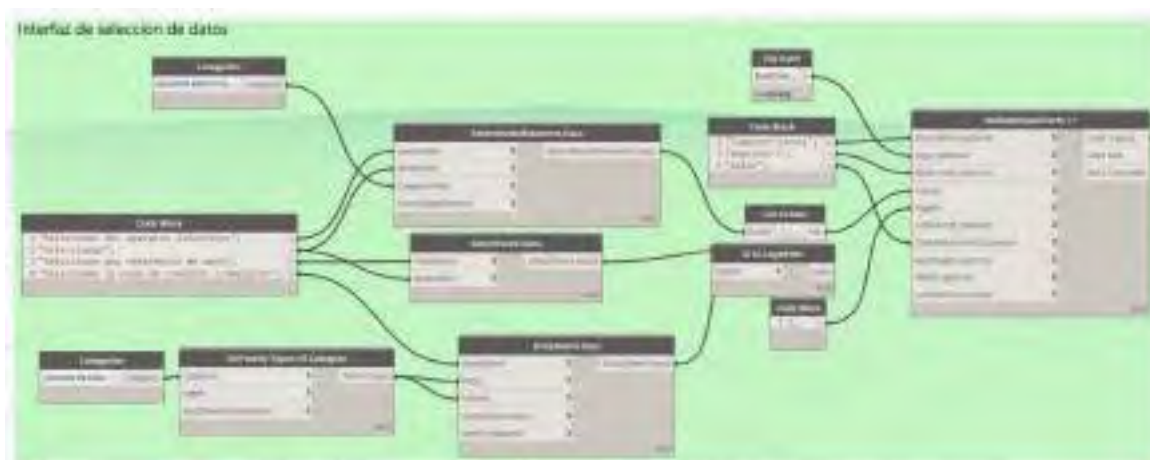
**5.3.6.2** Desarrollo del Script “Tomacorrientes”

La etapa de desarrollo del script “Tomacorrientes” se enfoca en reducir el sistema de nodos a dos etapas claramente diferenciadas, aprovechando los conocimientos previos de emplazamiento por diferentes parámetros de elementos en forma de puntos geométricos disponibles en la interfaz de programación de Dynamo.

En la primera etapa, se destaca la relevancia de la interfaz de entrada de datos, que se enfoca en la selección de aparatos eléctricos filtrados por categoría. Al igual que en el caso del script dedicado a la iluminación, el emplazamiento de elementos, cuando se trata de elementos de conexión como los

**Figura 5.32**

*Interfaz de selección de datos para el emplazamiento de cajas de conexión para tomacorrientes*



*Fuente:* Implementación propia.

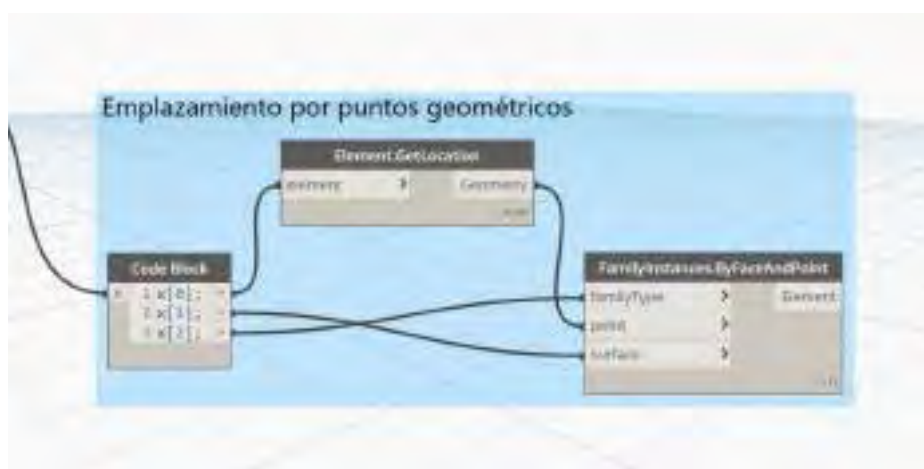
tomacorrientes, tiene menos probabilidad de fallo cuando se utiliza un tratamiento de emplazamiento por caras y luego por anfitrión, gestionado por Revit.

La Figura 5.32 muestra la interfaz de selección desarrollada para este script. En los objetivos programables, esta interfaz se amplía al permitir seleccionar el tipo de caja que se ha de ubicar. Esta funcionalidad es especialmente útil en proyectos complejos donde pueden existir diferentes tipos de cajas de conexiones.

Una vez que se tienen enlistados los datos de entrada básicos, el emplazamiento de los tomacorrientes se puede llevar a cabo con la información geométrica de las coordenadas *XYZ* que manejan tanto Revit como Dynamo. La Figura 5.33 muestra cómo se realiza este proceso.

**Figura 5.33**

*Instalación de cajas de conexión por puntos geométricos de tomacorrientes*



*Fuente:* Implementación propia.

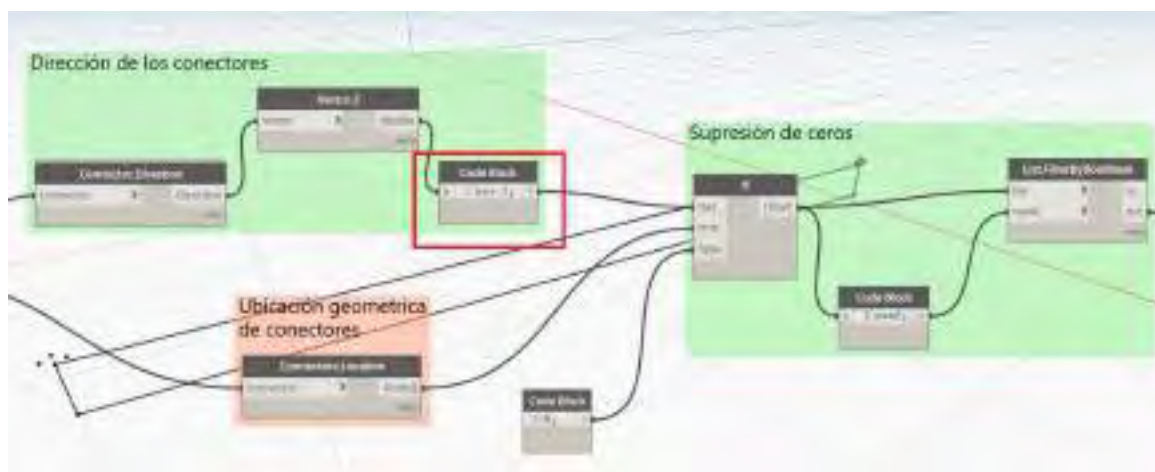
### 5.3.6.3 Desarrollo del Script “Conductos II”

Una vez obtenidas las ubicaciones de las cajas de conexión correspondientes a cada punto geométrico detectado, el siguiente paso es realizar el entubado por suelo para las instalaciones eléctricas generadas. Para este desarrollo, se ha planteado una metodología basada en el orden de niveles en el que se está ejecutando, lo que significa que al proponer una selección de elementos, estos deben estar ubicados en el mismo nivel.

En un enfoque similar al utilizado previamente para el sistema de iluminación, se emplea el poderoso sistema de nodos dentro de Dynamo para facilitar el proceso. Aprovechando el mismo sistema de selección múltiple y mapeado de conectores utilizado anteriormente (Figura 5.26), en esta ocasión se realiza una especificación adicional para identificar específicamente los conectores con un vector direccional  $-1$ , es decir, aquellos que se orientan hacia abajo en la dirección del eje Z (Ver Figura 5.34).

**Figura 5.34**

*Detección de conectores con vector direccional  $-1$  para sistemas de tomacorrientes*



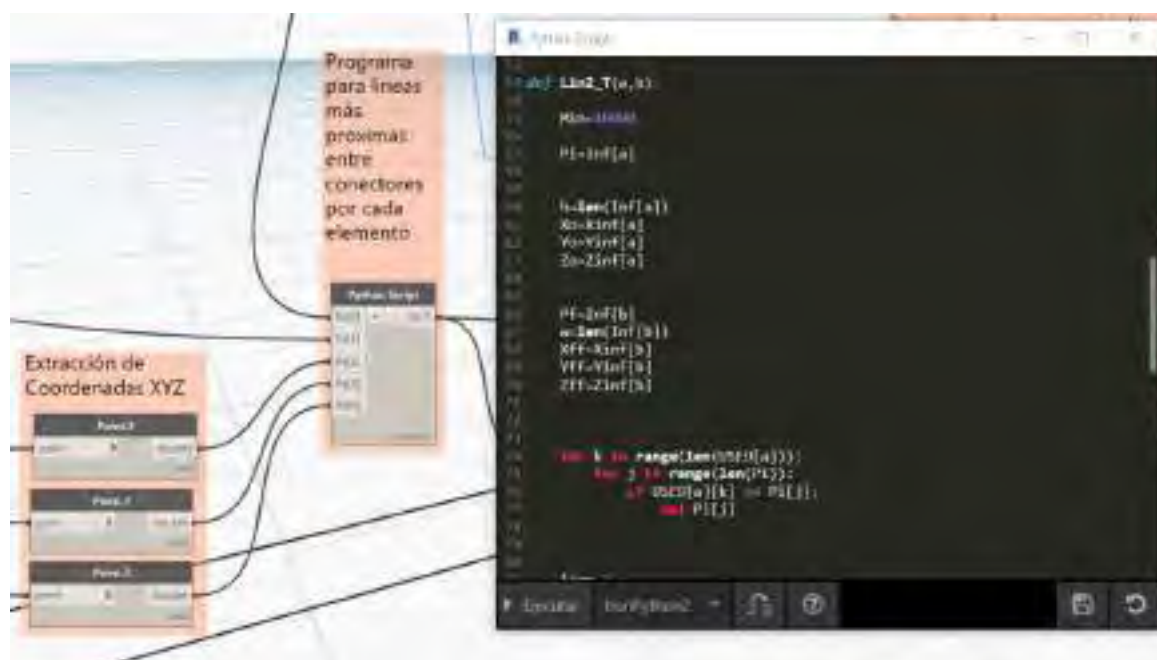
*Fuente:* Implementación propia.

Una vez que los conectores han sido mapeados correctamente, es factible llevar a cabo la nueva implementación de los algoritmos 5.7 y 5.8 mediante el uso de nodos que permiten la aplicación de programación tradicional dentro de Dynamo. Esto se puede apreciar en la Figura 5.35, y el código fuente completo se encuentra disponible en el Anexo A - Código A.4.

Es fundamental asegurarse de tomar la elevación precisa del nivel en el que se está trabajando, ya que esto es crucial para la correcta traslación de puntos dentro de las coordenadas Z. En el entorno de Revit, se puede obtener esta información utilizando el comando adecuado (*x.Elevation*), donde “x” representa la referencia de nivel del cual se desea extraer la elevación.

**Figura 5.35**

*Implementación de los algoritmos 5.7 y 5.8 en Dynamo*



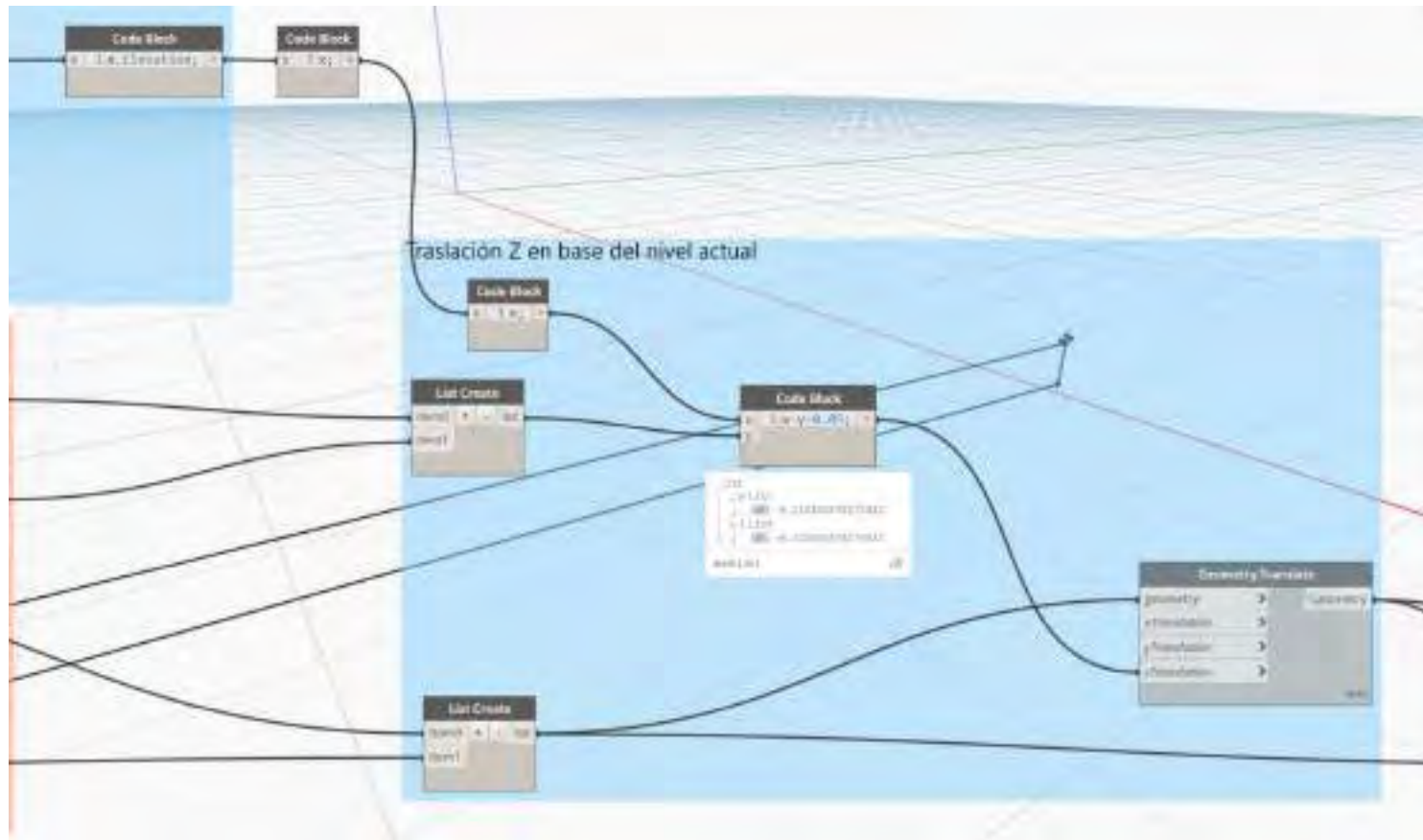
*Fuente:* Implementación propia.

Este proceso cobra especial importancia, ya que evita errores graves en el modelado BIM, como el diseño de conductos eléctricos que se solapan con elementos del suelo, lo cual puede afectar la integridad y funcionalidad del proyecto (Ver Figura 5.36).

Para finalizar, el proceso de toma de datos para reemplazar líneas geométricas por tuberías sigue la misma lógica presentada anteriormente en la Figura 5.29, tomando en cuenta el tipo de Familia que maneja Revit y el sistema de diámetros requeridos.

Figura 5.36

Traslación de puntos Z para sistemas de tomacorrientes



Fuente: Implementación propia.

### 5.3.7 Corriente Débil

La instalación de corriente débil en el ámbito interior es similar en ciertos aspectos a las instalaciones eléctricas convencionales. Implica la colocación de montantes y puntos específicos para satisfacer las necesidades del sistema. Sin embargo, a diferencia de las instalaciones eléctricas principales, las de corriente débil se enfocan en equipos que requieren un entubado básico y cajas de conexión adecuadas para alojar el cableado correspondiente a los circuitos necesarios.

El diseño de estas instalaciones puede variar en función de aspectos técnicos, criterios profesionales y normativas aplicables. En este sentido, la automatización total de este proceso requeriría un estudio especializado en el área.

Para los objetivos de esta tesis, se plantea adaptar los programas previamente desarrollados en Dynamo para la generación de conductos. Esto implica que los programas existentes para el entubado por techos, suelo y generación de montantes pueden ser reutilizados y adaptados para su uso en instalaciones de corriente débil. Estas herramientas pueden ser implementadas en una ficha adicional dentro del entorno “Electrical” de Revit, lo que facilitaría su reutilización y mantenimiento del orden en el proceso de diseño.

En resumen, si bien la automatización total de las instalaciones de corriente débil requiere de un enfoque especializado, se pueden aprovechar y adaptar los programas ya desarrollados en Dynamo para optimizar y agilizar ciertas tareas relacionadas con el entubado y generación de montantes. Esto proporciona una solución práctica y eficiente para el diseño de instalaciones de corriente débil dentro de proyectos BIM.

## 5.4 Cálculo

El cálculo de instalaciones eléctricas interiores comprende diversos procesos de aplicación matemática y física que proporcionan una visión proyectual de la carga que se está conectando, así como de la corriente eléctrica que fluye por los circuitos planteados. Esto permite prever dispositivos que cumplan roles de seguridad para la instalación eléctrica y calcular las caídas de tensión causadas por la corriente que está siendo calculada. Además, teniendo en cuenta las longitudes de los circuitos y sus secciones, se asegura el correcto funcionamiento de la instalación.

Tanto en los flujos de trabajo tradicionales como en los nuevos basados en BIM, es esencial recurrir a estos cálculos para calificar el diseño generado en BIM, ya que esta etapa garantiza el cumplimiento de los aspectos normativos y, por ende, la característica BIM de “Gestión”.

En este contexto, la correcta aplicación de los cálculos está estrechamente relacionada con el manejo informativo BIM que se puede lograr. Se hace referencia al *Workflow* de Chen et al. visto

anteriormente en las figuras Figura 3.3 y Figura 3.4. Este aspecto ha sido previsto con el desarrollo de herramientas informáticas en el ámbito de modelado, como se describió en la sección anterior. En el cálculo eléctrico BIM, que implica aspectos normativos, se lleva a cabo el estudio de procesos de *Code Checking* específicos para el proyecto.

El primer marco técnico y normativo que respalda los diseños de instalaciones eléctricas en territorio peruano es el Código Nacional de Electricidad (CNE), el cual es un elemento principal que respalda muchos de los alcances propuestos por el Reglamento Nacional de Edificaciones (RNE). En consecuencia, la base algorítmica planteada para la ejecución de los cálculos automatizados debe cumplir con estos referentes.

#### 5.4.1 *Base del Algoritmo*

Para fines de la ejecución del cálculo de instalaciones eléctricas interiores, las herramientas a programar se dirigen hacia los principales objetos de cálculo expuestos anteriormente como lo son:

- Cálculo de Cargas o Cuadro de Cargas
- Cálculo de Caídas de Tensión

##### 5.4.1.1 Cálculo de Cargas

Para obtener datos que sirvan como parámetros de entrada en los nuevos cálculos, es prioritario calcular la carga proyectada por edificación. En esta base programable, se otorga gran importancia al cumplimiento normativo. Además de seguir las indicaciones claras establecidas en el CNE-U, se busca aumentar el potencial de esta meta programable al considerar el “Manual de Sustentación del Código Nacional de Electricidad Utilización 2006” (Ministerio de Energía y Minas, 2008).

En el mencionado manual, en los apuntes para la *Sección 050* del CNE-U, se encuentra un conjunto de pasos para el cálculo de cargas. Estos pasos pueden entenderse, en términos generales, como un algoritmo detallado que incluye ejemplos y un formato que puede servir de base para la exposición cuantitativa de los cálculos realizados.

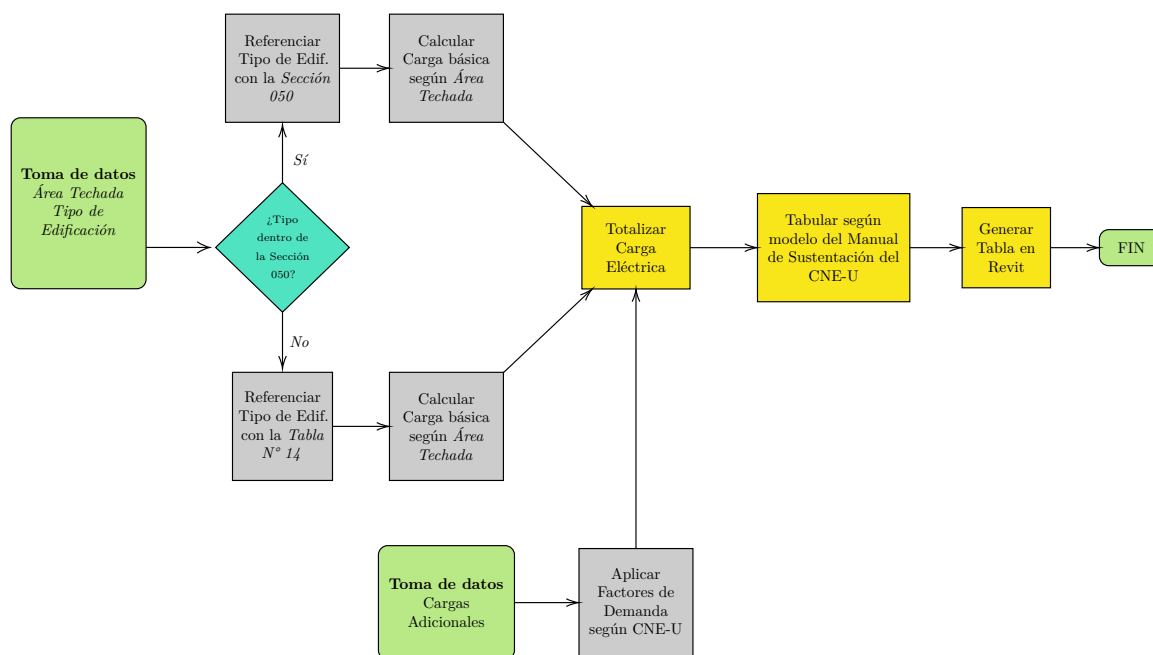
La Figura 5.37 muestra un diagrama de flujo que sirve como referencia en el proceso de programación para el cálculo de cargas.

Encontrando nodos dentro de *Dynamo* que permiten la manipulación de tablas dentro de Revit, como se mostró en la Figura 4.9 (ver pag. 104), es importante comprender que las tablas dentro del software BIM no pueden ser llenadas de forma manual como se haría en una interfaz común, como por ejemplo, *Microsoft Excel*.



Figura 5.37

Diagrama de Flujo para el cálculo de cargas



Fuente: Elaboración propia.

Dentro de este contexto, si un cuadro de cargas, siguiendo las indicaciones del ya mencionado “Manual de Sustentación del Código Nacional de Electricidad Utilización 2006”, incluye fórmulas, datos y descripciones que no pueden ser incorporados de manera directa en Revit, es necesario implementar la funcionalidad correspondiente en un formato básico.

La Tabla 5.1 muestra una guía referencial con los encabezados que debe tener el cuadro de cargas generado en Revit. Además, teniendo en cuenta la eventual interacción con lenguajes de programación, es convencional utilizar listas con números de orden que comienzan siempre por cero.

Tabla 5.1

Campos de un Cuadro de Cargas según Ministerio de Energía y Minas (2008)

<i>Campo</i>	Norma	Carga	Descripción	Potencia	Factor de Demanda	Demanda Máxima
<i>N° orden programable</i>	0	1	2	3	4	5

Fuente: Elaboración propia.

Después de comprender los datos que la tabla debe contener, el siguiente paso en la propuesta del algoritmo es evaluar la lógica que debe tener cada cálculo diferenciado, es decir, en el CNE-U, dentro de la respectiva *Sección 050*, se tienen diferenciados los siguientes tipos de edificación:

- Vivienda Unifamiliar

- Edificios Departamentales
- Escuelas
- Hospitales
- Hoteles
- Otros (Referencia de uso hacia la Tabla 14 del CNE-U)

Tras una evaluación del documento de referencia, se puede observar una lógica similar entre los distintos tipos de edificación para el cálculo de las cargas.

Por ejemplo, tanto en Viviendas Unifamiliares como en Edificios Departamentales, el cálculo de la carga básica se realiza a través de “Fraccionamientos” de  $90m^2$  y  $45m^2$ , respectivamente. En cambio, para los otros tipos de edificación, se indica el uso de un valor de “Potencia Superficial” ( $W/m^2$ ), lo que añade complejidad debido a que en Escuelas y Hospitales se debe diferenciar el uso de ciertas áreas de la edificación. En estos casos, es posible recalculer la carga básica haciendo uso de los factores proporcionados por el CNE-U.

Además del cálculo de la Carga Básica por Edificación, se debe considerar la carga adicional, ya que dependiendo del tipo de edificación, esta puede depender de la presencia o ausencia de aparatos especializados, como una cocina eléctrica en el caso de Viviendas Unifamiliares y Departamentos.

Otro tipo de carga especializada a considerar para todos los casos son las Cargas de Calefacción y de Aire Acondicionado, para las cuales es necesario recurrir al criterio normativo propuesto por el CNE-U, ya que describe los factores de demanda específicos en su respectiva *Sección 270*.

La Tabla 5.2 muestra un resumen de la propuesta para el desarrollo del módulo correspondiente a la generación del cuadro de cargas dentro del software BIM, donde se distribuirán las instrucciones programables dentro de la interfaz de Dynamo.

**Tabla 5.2**

*Resumen de consideraciones para el cálculo de cargas según CNE-U*

<i>Tipo de Edificación</i>	<i>Carga Básica</i>	<i>Carga Adicional</i>
Vivienda Unifamiliar	Cálculo por fraccionamiento	Dependiente de cocina eléctrica
Edificio Departamental		
Escuelas	Cálculo por Potencia superficial	Sumado para la aplicación de factores de demanda
Hospitales		
Hoteles		
Otros	Cálculo por Potencia Superficial (Tabla 14)	Junto a Factores de demanda sustentados

*Fuente:* Elaboración propia.

Es comprensible que el cálculo por Potencia Superficial pueda lograrse mediante una multiplicación simple con un dato de entrada, lo que lo hace relativamente sencillo de implementar en el módulo. Sin embargo, la dificultad radica en el planteamiento del Cálculo por Fraccionamiento, ya que, considerando la posibilidad de tener datos de área muy grandes, se requerirá implementar procesos programables de tipo repetitivos.

Esto evidencia una vez más la necesidad de recurrir a un estilo de programación tradicional para manejar el cálculo de fraccionamientos, ya que la repetición de cálculos por área puede ser compleja de lograr en entornos BIM y puede requerir estructuras de control más avanzadas (Ver Algoritmo 5.9).

---

**Algoritmo 5.9** Cálculo de Carga Básica para Viviendas Unifamiliares y Edificios Departamentales

---

**Require:** Área Techada, Fracciones de Área, Potencia por Fracción

```

1:  $PrimA \leftarrow$  Primera Fracción de Área
2:  $SegA \leftarrow$  Segunda Fracción de Área
3:  $FracA \leftarrow$  Fracción Restante de Área
4:  $PrimP \leftarrow$  Potencia por Primera Fracción de Área
5:  $SegP \leftarrow$  Potencia por Segunda Fracción de Área
6:  $FracP \leftarrow$  Potencia por Fracción Restante de Área
7:  $Pot \leftarrow [0, 0, 0]$ 
8: for  $i = 1 \dots 3$  do
9:   if  $i = 0$  then
10:     $A \leftarrow A - PrimA$ 
11:     $Pot[i] \leftarrow PrimP$ 
12:   else if  $i = 1$  and  $A > 0$  then
13:     $A \leftarrow A - SegA$ 
14:     $Pot[i] \leftarrow SegP$ 
15:   else
16:    while  $A > 0$  do
17:      $A \leftarrow A - FracA$ 
18:      $Pot[i] \leftarrow Pot[i] + FracP$ 
19:    end while
20:   end if
21: end for
   return  $Pot$ 

```

*Fuente:* Elaboración propia.

---

Considerando lo explicado anteriormente en la Tabla 5.1, y con el objetivo de reducir el esfuerzo computacional al incluir numerosos nodos, se propone utilizar el Algoritmo 5.10. Este último define su funcionamiento en la escritura automática de los nombres de las cargas y sus respectivas potencias, aprovechando la información BIM extraída del proyecto.

El planteamiento de las cargas adicionales se puede realizar a través de la misma interfaz de selección de elementos BIM, como artefactos eléctricos, elementos de iluminación especializada u otros componentes del proyecto. Sin embargo, en casos especiales como Viviendas Unifamiliares y Edificios Departamentales, donde la presencia de una cocina eléctrica debe ser considerada cuidadosamente, el uso de nodos en Dynamo para aplicar nuevos factores de demanda puede no ser suficiente.

**Algoritmo 5.10** Creación de Matriz de Datos para Cargas Adicionales**Require:** Listas: Datos de Potencia de cada carga, Nombre de cada carga**Require:** Cadena Texto: Norma del CNE-U a la que hace referencia

```

1:  $PotAd \leftarrow$  Potencias de cada Carga
2:  $NombAd \leftarrow$  Nombre de cada Carga
3:  $Norm \leftarrow$  Norma Referencial
4:  $TamVert \leftarrow$  Tamaño del Vector  $PotAd$ 
5:  $M \leftarrow$  Matriz vacía de orden  $6X(TamVert + 1)$ 
6: for  $i = 1 \dots (TamVert + 1)$  do
7:   for  $j = 1 \dots 6$  do
8:     if  $i = 0$  and  $j = 0$  then
9:        $M[i, j] \leftarrow Norm$ 
10:    else if  $j = 1$  and  $i = 0$  then
11:       $M[i, j] \leftarrow$  "Carga Adicional"
12:    else if  $i = 1$  and  $j \neq 0$  then
13:       $M[i, j] \leftarrow NombAd[i - 1]$ 
14:    else if  $j = 2$  then
15:       $M[i, j] \leftarrow$  Cadena Texto Vacía
16:    else if  $j = 3$  and  $i > 0$  then
17:       $M[i, j] \leftarrow PotAd[i - 1]$ 
18:    else if  $j = 4$  then
19:       $M[i, j] \leftarrow$  Cadena Texto Vacía
20:    else if  $j = 5$  and  $i > 0$  then
21:       $M[i, j] \leftarrow PotAd[i - 1]$ 
22:    end if
23:  end for
24: end for
    return  $M$ 

```

*Fuente:* Elaboración propia.

Para evitar posibles correcciones erróneas en la potencia debido a la presencia de una cocina eléctrica, es necesario implementar líneas de código tradicional para evaluar escenarios específicos y aplicar la corrección de la carga en función de la presencia o ausencia de la cocina eléctrica. Por este motivo, en el Algoritmo 5.11 se propone un programa corto y sencillo que evalúe de manera rápida y precisa la corrección de la carga según la presencia de una cocina eléctrica en la instalación eléctrica diseñada.

Dynamo se enfoca en el manejo de listas, y este algoritmo devuelve un resultado en forma de lista con cada potencia corregida. La ausencia del manejo de listas adicionales con nombres se debe a la posibilidad de aplicar el Algoritmo 5.10, con la única diferencia de cambiar el parámetro de entrada respecto a la potencia de cada carga adicional en casos excepcionales. De esta manera, ambos algoritmos pueden complementarse eficientemente para obtener los resultados deseados.

---

**Algoritmo 5.11** Cálculo de Cargas Adicionales por presencia de cocinas eléctricas
 

---

**Require:** Lista: Datos de Potencia de cada carga adicional

**Require:** Valor Booleano: Presencia o no de una cocina eléctrica

```

1: NoCocEle ← Presencia o no de una cocina eléctrica (Booleano)
2: Pot_Ad ← Lista con potencia de cada carga adicional
3: Tam ← Tamaño del Vector Pot_Ad
4: N_Pot ← Lista de ceros de tamaño Tam
5: for  $i = 1 \dots TamVert$  do
6:   if NoCocEle ≠ 0 then
7:     if Pot_Ad[ $i$ ] > 6000 then
8:       N_Pot[ $i$ ] ← Pot_Ad[ $i$ ] + (Pot_Ad[ $i$ ] - 6000) * 0,25
9:     else
10:      N_Pot[ $i$ ] = Pot_Ad[ $i$ ]
11:    end if
12:  else
13:    N_Pot[ $i$ ] ← Pot_Ad[ $i$ ] * 0,25
14:  end if
15: end for
    return N_Pot
  
```

*Fuente:* Elaboración propia.

---

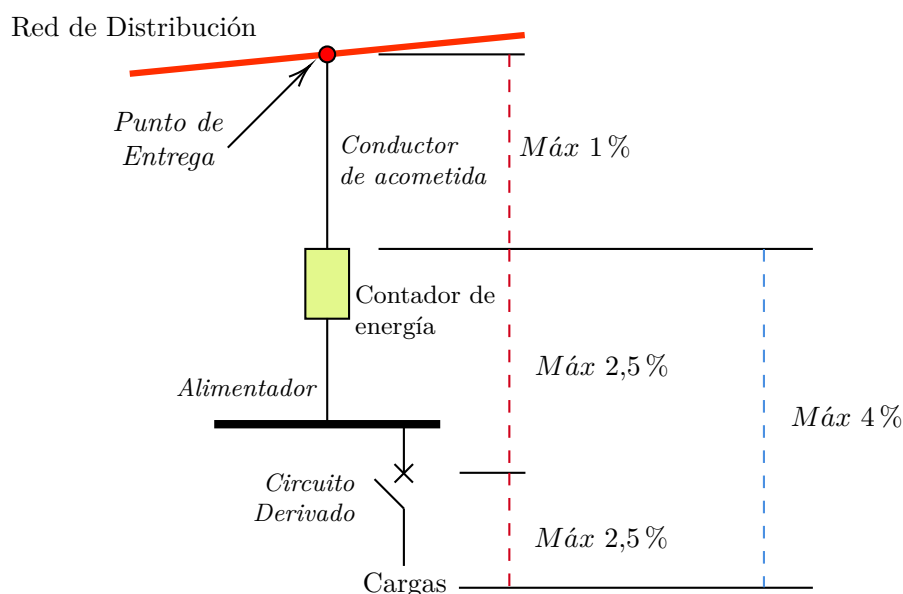
### 5.4.1.2 Caídas de Tensión

El cálculo de caídas de tensión es crucial para asegurar un funcionamiento óptimo de la instalación eléctrica. Un diseño bien fundamentado con base en los alcances BIM ofrece una visión más clara del rendimiento esperado y permite un cálculo más preciso y eficiente, al aprovechar parámetros de diseño como longitud, material y sección del conductor, entre otros.

Actualmente, el CNE-U, en la *Sección 050-102*, proporciona tolerancias máximas como límites para la caída de tensión, tanto para el conjunto de alimentadores y circuitos derivados, como para circuitos derivados individuales en puntos de conexión, como tableros de distribución. La Figura 5.38 resume de manera visual los alcances normativos proporcionados por el CNE-U.

**Figura 5.38**

*Máximas caídas de tensión permitidas en un circuito*



*Fuente:* Elaboración propia adaptado de Ministerio de Energía y Minas (2008, p. 48).

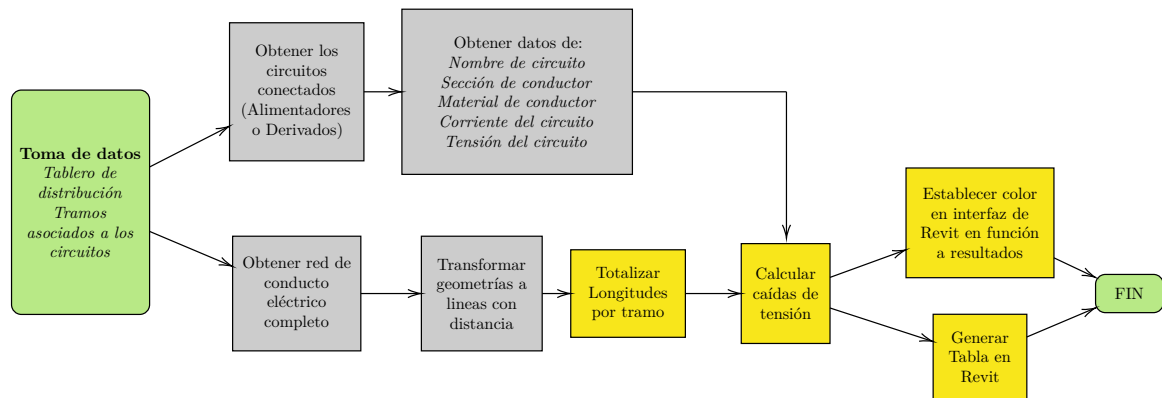
Dado este escenario, se propone utilizar un límite de cálculo del 2,5% para tanto circuitos alimentadores como circuitos derivados, según lo establecido por el CNE-U. En este caso, la Figura 5.39 muestra el diagrama de flujo sobre el cual se llevarán a cabo las tareas de programación para el cálculo de caídas de tensión.

En el escenario donde se tienen múltiples elementos organizados en listas y sublistas, se propone recurrir a la programación tradicional para ejecutar los cálculos de caídas de tensión, aprovechando el multinivel de los datos, como Tensiones y Materiales de los conductores, entre otros.

La ventaja de implementar líneas de programación es que al computar los datos de entrada, es posible generar una lista que pueda ser utilizada en un cuadro de Revit, similar a la base programable

**Figura 5.39**

*Diagrama de flujo para cálculo de caídas de tensión*



*Fuente:* Elaboración propia.

para la generación del cuadro de cargas mencionado anteriormente. Este aspecto se puede apreciar en el Algoritmo 5.12.

---

**Algoritmo 5.12** Cálculo de Caídas de Tensión y Generación de Matriz de resultados

**Require:** Lista: Datos de Nombres de circuitos, Tensión, Número de polos, Longitud del tramo,

Material del conductor, Corriente, Sección del conductor

```

1:  $Cir \leftarrow$  Nombre del Circuito
2:  $V \leftarrow$  Tensión del circuito
3:  $NP \leftarrow$  Número de polos
4:  $Long \leftarrow$  Longitud del tramo asociado al circuito
5:  $Mat \leftarrow$  Material del conductor
6:  $I \leftarrow$  Corriente del circuito
7:  $S \leftarrow$  Sección del conductor
8:  $tam \leftarrow$  Tamaño del vector de número de circuitos
9:  $M \leftarrow$  Matriz vacía de orden  $6X(tam)$ 
10: for  $i = 1 \dots tam$  do
11:   if  $NP[i] = 1$  then
12:      $k \leftarrow 2$ 
13:   else if  $NP[i] = 3$  then
14:      $k \leftarrow \sqrt{3}$ 
15:   end if
16:   if  $Mat[i] = \text{"Cobre"}$  then
17:      $p \leftarrow 0,0176$ 
18:   else if  $Mat[i] = \text{"Aluminio"}$  then
19:      $p \leftarrow 0,0260$ 
20:   end if
21:    $DV \leftarrow k * I[i] * 1,25 * \frac{Long[i]}{S[i]} * 0,9 * p$ 
22:    $M[i][0] \leftarrow Cir[i]$ 
23:    $M[i][1] \leftarrow I[i] * 1,25$ 
24:    $M[i][2] \leftarrow Long[i]$ 
25:    $M[i][3] \leftarrow Mat[i]$ 
26:    $M[i][4] \leftarrow S[i]$ 
27:    $M[i][5] \leftarrow DV$ 
28: end for
   return  $M$ 

```

*Fuente:* Elaboración propia.

---



### 5.4.2 Desarrollo del Script “Cargas”

El desarrollo del script “Cargas” se basa en la primera meta programable del cálculo de instalaciones eléctricas, que es la correcta generación del Cuadro de Cargas, como se mostró en el numeral anterior. Para este script, se tomará en cuenta el diagrama de flujo descrito en la Figura 5.37, así como los algoritmos 5.9, 5.10 y 5.11.

Al igual que en casos anteriores, para la ejecución y desarrollo dentro de Dynamo, se utilizará paquetería adicional, que incluye:

- Paquetes Adicionales

1. *archi-lab* por Konrad K Sobon<sup>7</sup>
2. *Data-Shapes* por Elayoubi Mostafa
3. *Orchid* por erfajo
4. *DynaMEP* por Chuong Ho<sup>8</sup>
5. *Rhythm* por John Pierson<sup>9</sup>

Con los nodos adecuados, se puede establecer una interfaz utilizable en Revit que permita tomar datos de áreas, diferenciando claramente la posibilidad de recurrir al uso de la Tabla 14 del CNE-U. Esta etapa inicial del desarrollo es de gran importancia, ya que se debe diferenciar si la toma de datos de área será única o diferenciada, como en el caso de Escuelas y Hospitales. Por esta razón, en la Figura 5.40 se propone un flujo de datos acorde a la normativa vigente.

Dado que existen múltiples formas de planificar áreas dentro de Dynamo y que, comúnmente, la normativa correspondiente varía en términos de dimensiones exteriores o interiores, la siguiente etapa en la toma de datos consiste en introducir parámetros numéricos directamente, respetando las unidades correspondientes ( $m^2$ ). Sin embargo, debido a la existencia de diferentes tipos de áreas, como se mencionó en el párrafo anterior, es necesario filtrar qué tipo de área se toma en función de la selección previa del Tipo de Edificación.

---

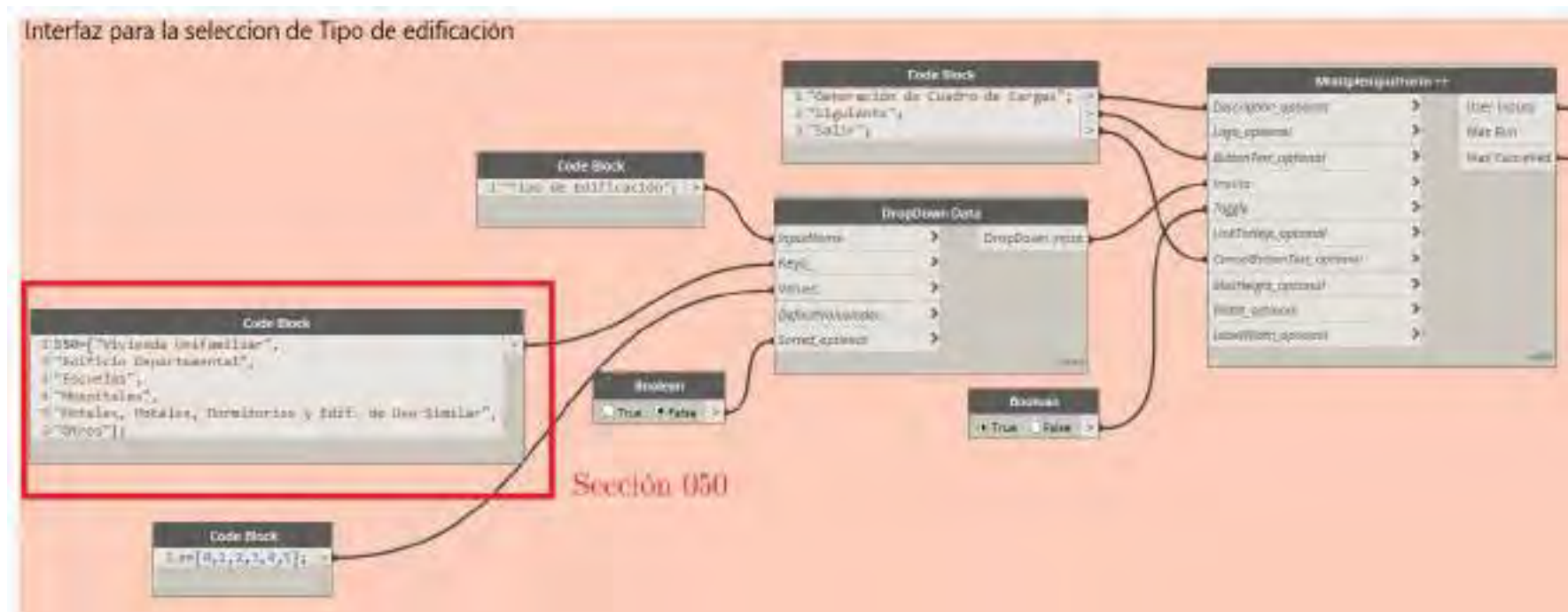
<sup>7</sup> Sitio Web el autor: <https://archi-lab.net>

<sup>8</sup> Consulta en Repositorio oficial de Dynamo: <https://dynamopackages.com/>

<sup>9</sup> Repositorio del autor: <https://github.com/johnpierson/RhythmForDynamo>

**Figura 5.40**

*Interfaz de selección de Tipo de Edificación según alcances del CNE-U*

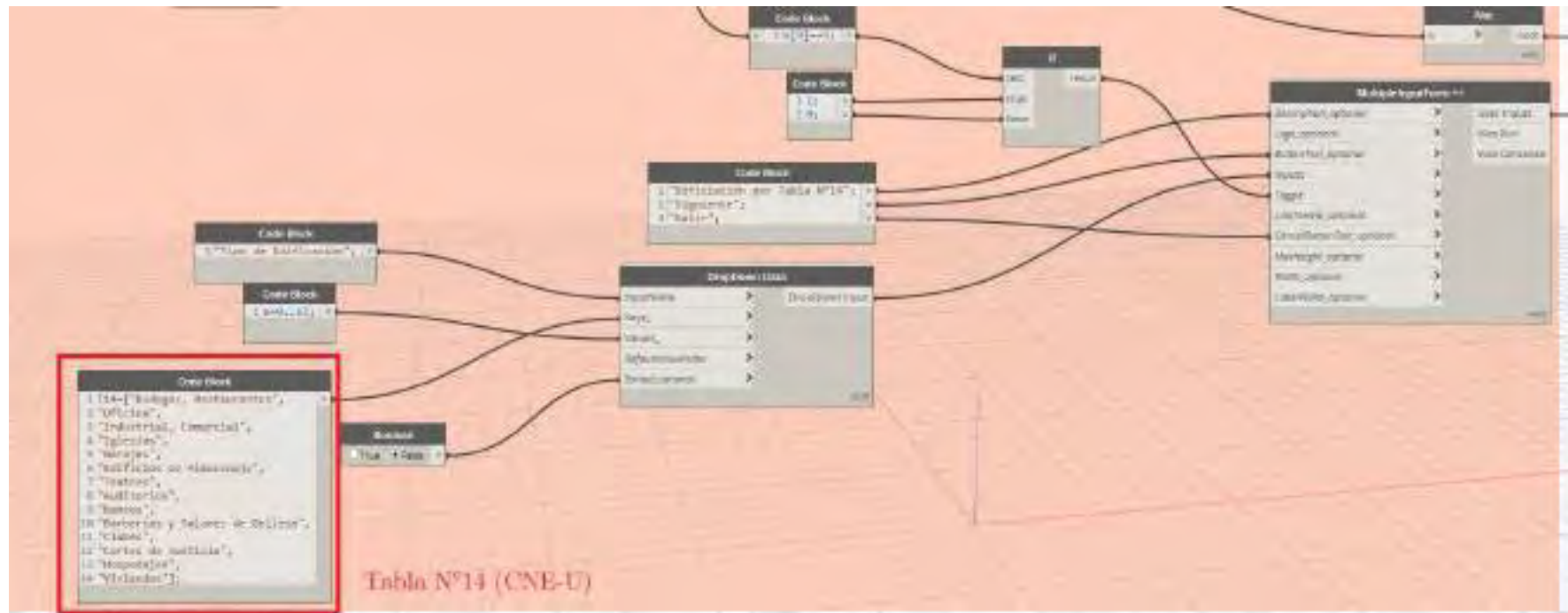


(a) Primera parte del script

Fuente: Implementación propia.

Figura 5.40

Interfaz de selección de Tipo de Edificación según alcances del CNE-U (Continuación)



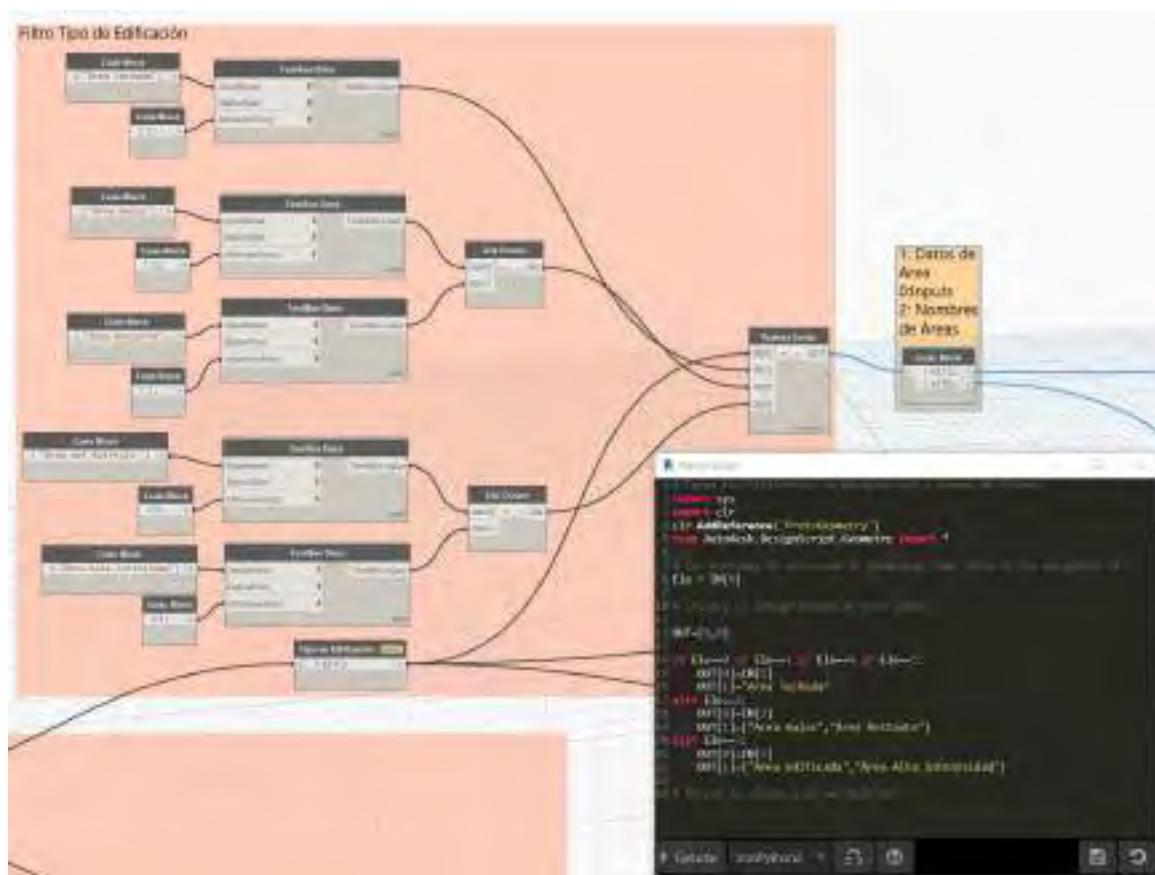
(b) Segunda parte del script

Fuente: Implementación propia.

La Figura 5.41 muestra un conjunto de nodos de selección, y con la ayuda de líneas programadas en una ventana Python, se permitirá únicamente la interacción con la entrada de datos de área según la elección realizada en la etapa anterior. De esta manera, se asegura que solo se ingresen los datos de área correspondientes al tipo de edificación seleccionado previamente, evitando posibles errores y garantizando la correcta asignación de las áreas en el cálculo de cargas.

**Figura 5.41**

*Filtro para áreas según selección del tipo de edificación*



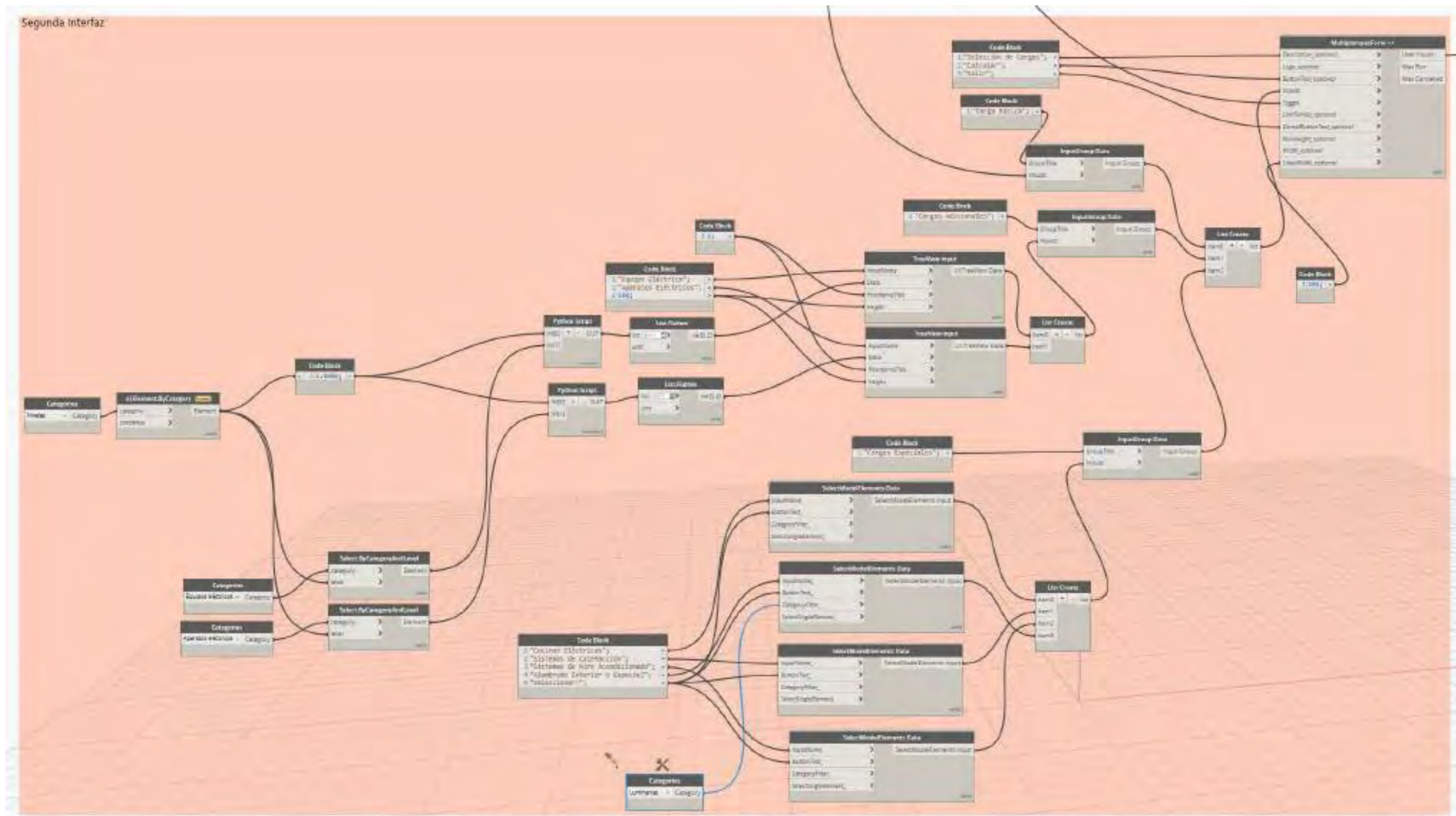
*Fuente:* Implementación propia.

De esta manera, con el resultado obtenido, se puede completar la interfaz de datos referentes al área del proyecto para el cálculo de la Carga Básica. Asimismo, en la misma ventana de entrada de datos, se puede aprovechar la funcionalidad para seleccionar las cargas existentes en el proyecto sobre las cuales se desea realizar el cálculo.

Es importante mencionar que se ha implementado un ordenamiento de las posibilidades, utilizando el filtro “Por Nivel” para las cargas adicionales. Además, se ha ampliado el alcance de esta interfaz para la selección directa de cargas que califiquen como Iluminación especializada, Aire Acondicionado, Calefacción y Cocinas Eléctricas (Figura 5.42). De esta manera, se facilita la selección de las cargas específicas que se deseen incluir en el cálculo de cargas eléctricas.

Figura 5.42

Interfaz para la toma de áreas y cargas eléctricas adicionales



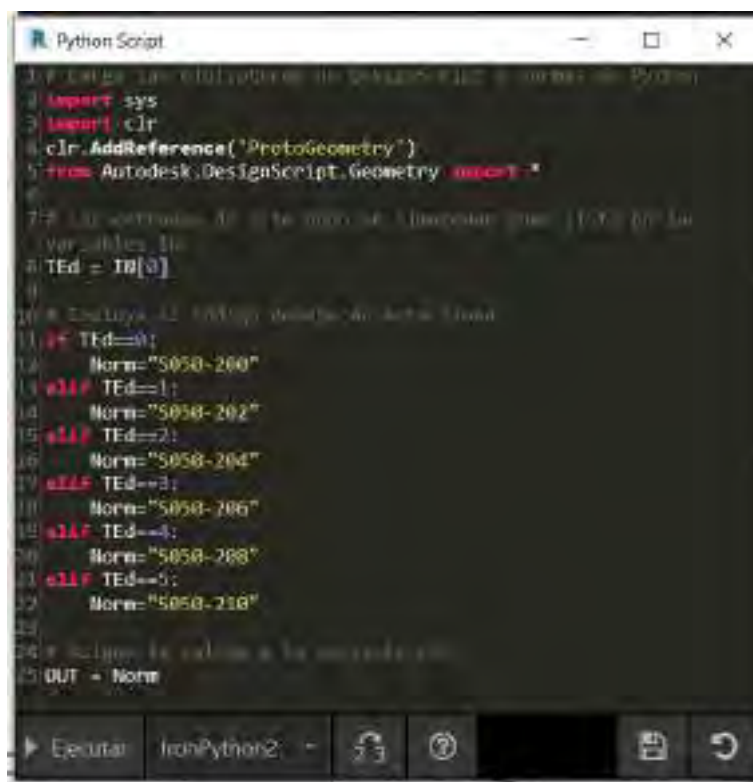
Fuente: Implementación propia.

Es fundamental reorganizar y derivar los datos obtenidos de las interfaces desarrolladas en las figuras 5.40 y 5.42 hacia el bloque de ejecución correspondiente, ya que, de no hacerlo, existe la posibilidad de que todo el flujo de programación colapse y no se obtengan resultados correctos.

Una alternativa encontrada para diferenciar los diferentes tipos de cálculo de carga básica, según la elección del tipo de edificación, fue implementar otro programa Python (Figura 5.43). Además, se incluye la norma a la cual hacer referencia en todos los cálculos a realizar. Aunque este aspecto pueda parecer innecesario debido a que el dato de entrada ya contiene un índice, la adición en mención evita el llamado repetitivo de los mismos nodos y logra un proceso de programación mucho más limpio y eficiente.

**Figura 5.43**

*Adición normativa del CNE-U a la toma de datos por tipo de edificación.*



```
Python Script
1 import sys
2 import clr
3 clr.AddReference("ProtoGeometry")
4 from Autodesk.DesignScript.Geometry import *
5
6 # Variables
7 TEd = TB[0]
8
9 # Estructura de datos para el cálculo de carga básica
10 TEd=0:
11     Norm="S050-200"
12 a11f TEd=1:
13     Norm="S050-202"
14 a11f TEd=2:
15     Norm="S050-204"
16 a11f TEd=3:
17     Norm="S050-206"
18 a11f TEd=4:
19     Norm="S050-208"
20 a11f TEd=5:
21     Norm="S050-210"
22
23
24 # Salida de datos a la consola
25 OUT = Norm
```

*Fuente:* Implementación propia.

En vista de esta necesidad conjunta, también se propone un ordenamiento básico de las cargas eléctricas que puedan ser seleccionadas, para facilitar su uso posterior en otros nodos de cálculo donde se requiera su inclusión en el cuadro de cargas o su recálculo mediante factores de demanda.

La ?? muestra un proceso donde se aplica la misma lógica de ordenamiento tanto para las cargas adicionales generalizadas como para las de tipo especial. Esta última categoría se extiende hacia una extracción adicional, en la que se separan las cargas categorizadas:

- Cocinas Eléctricas
- Calefacción
- Aire Acondicionado
- Iluminación Especializada

Con los datos de entrada, como el tipo de edificación y la carga adicional ordenada y categorizada, el programa cumple con el desarrollo de su objetivo programable para diferenciar entre los distintos tipos de cálculos mencionados en la Tabla 5.2.

Para los casos de Viviendas Unifamiliares y Edificios Departamentales, en la ?? se muestra el conjunto de nodos que permiten el cálculo de la carga básica en función del área techada, implementando el Algoritmo 5.9 a través de un nodo Python cuyo código se detalla en el Anexo A - Código A.5.

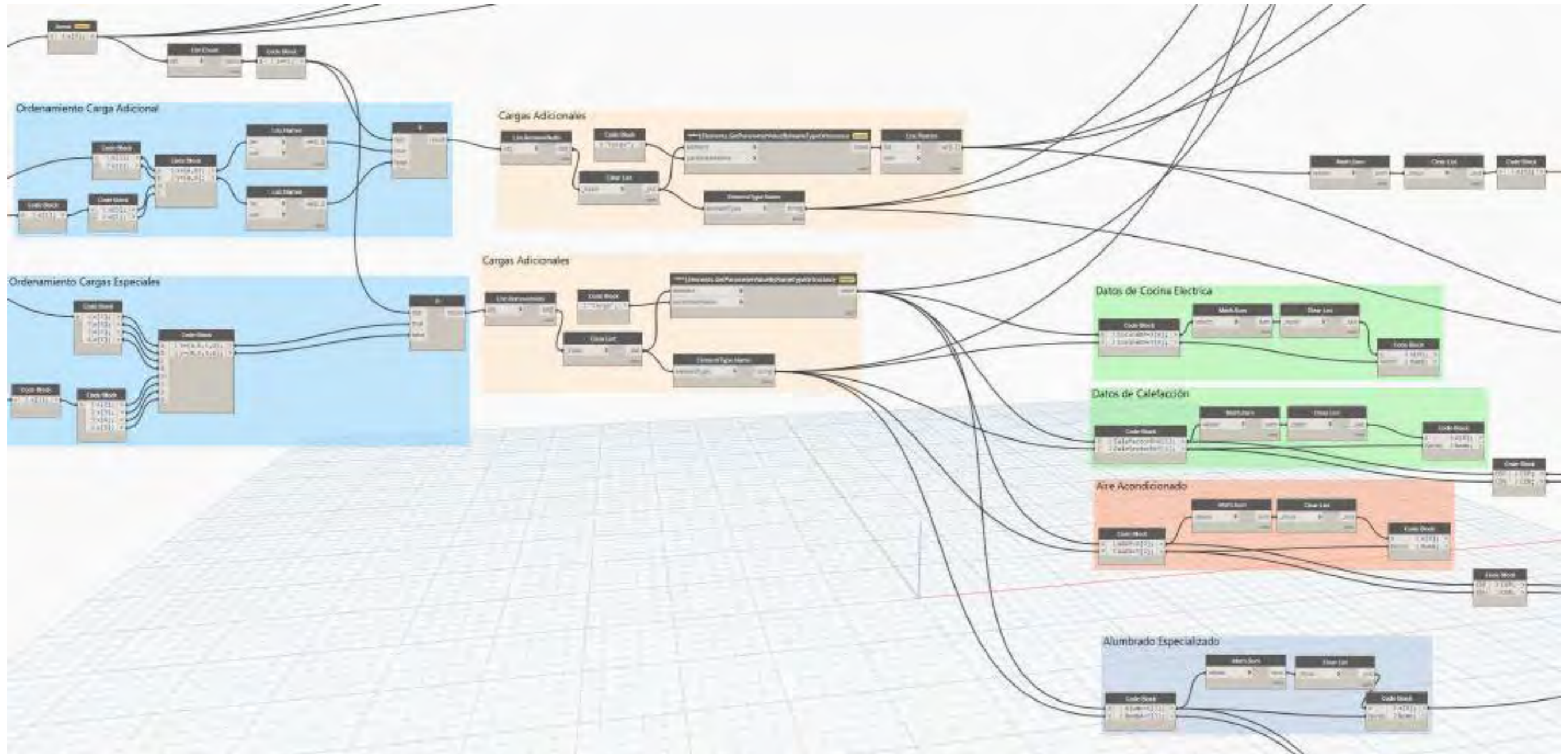
La complejidad de estos casos requiere también la inclusión del cálculo de la carga adicional, teniendo en cuenta las indicaciones del CNE-U respecto a la presencia o ausencia de una cocina eléctrica. Además, la norma sugiere el uso de factores de demanda únicos para cargas de calefacción de uso doméstico en comparación con usos comerciales o comunes.

En la ??, se muestran ambos procesos descritos anteriormente, junto con el Algoritmo 5.11.

Los códigos fuente Python para los Algoritmos 5.9 y 5.11 se pueden consultar en el Anexo A - Código A.5 y Código A.6 respectivamente.

Figura 5.44

Ordenamiento de Cargas Eléctricas Adicionales y de tipo Especial



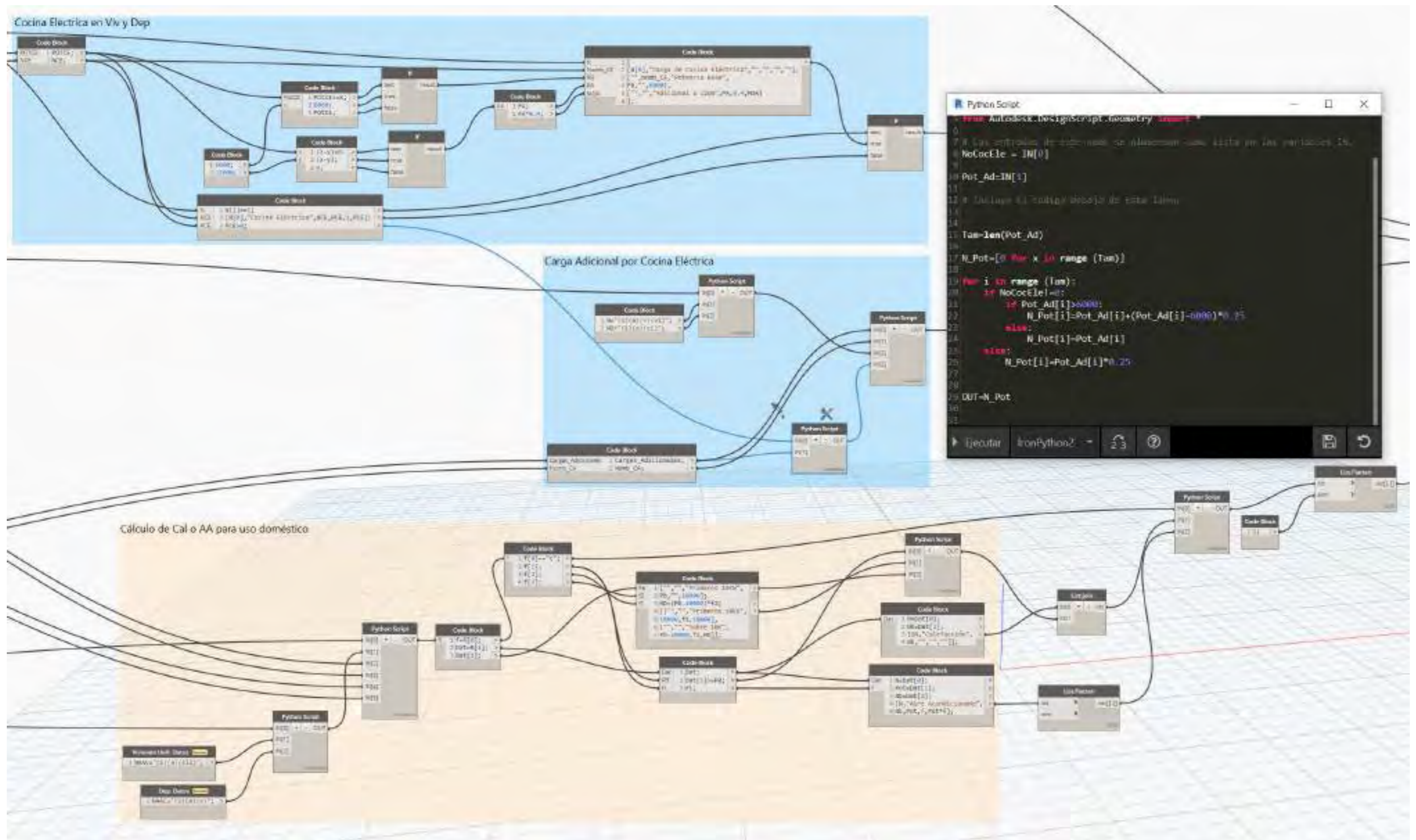
Fuente: Implementación propia.





Figura 5.46

Cálculo de Carga Adicional, Cocina Eléctrica y Aire Acondicionado o Calefacción para Viviendas Unifamiliares y Departamentos en Dynamo



Fuente: Implementación propia.

Ya que se ha diferenciado el cálculo de carga eléctrica tanto por fraccionamiento como por potencia superficial. En el caso del cálculo por potencia superficial, se han distinguido las edificaciones de tipo Escuelas, Hospitales y Hoteles. Sin embargo, solo en los dos primeros casos se diferencian espacios con un uso general y otro especializado, mientras que para Hoteles se considera únicamente un tipo de área (Ver Tabla 5.3).

Esta distinción resulta crucial para asegurar la precisión y adecuación de los cálculos a las necesidades específicas de cada tipo de edificación. Al considerar las diferencias en los usos y requerimientos eléctricos, se logra una mayor exactitud en el cálculo de la carga eléctrica para cada proyecto.

**Tabla 5.3**

*Tipos de Área para el cálculo por potencia superficial ( $W/m^2$ )*

<i>Tipo de Edificación</i>	<i>Áreas</i>
Hospitales	Área de la Edificación
	Área de Alta Intensidad
Escuelas	Área de Aulas
	Área restante
Hoteles	Área de la Edificación

*Fuente:* Elaboración propia.

Se propone un enfoque de menor esfuerzo computacional en el desarrollo del presente script al permitir el mismo procedimiento de cálculo para Escuelas y Hospitales. Dado que ambos tipos de edificación tienen dos áreas diferenciables, se puede utilizar una lógica de programación similar, y también es necesario agregar una sumatoria de las cargas adicionales para poder recalcularlas junto con la carga básica mediante los factores de demanda determinados por el CNE-U.

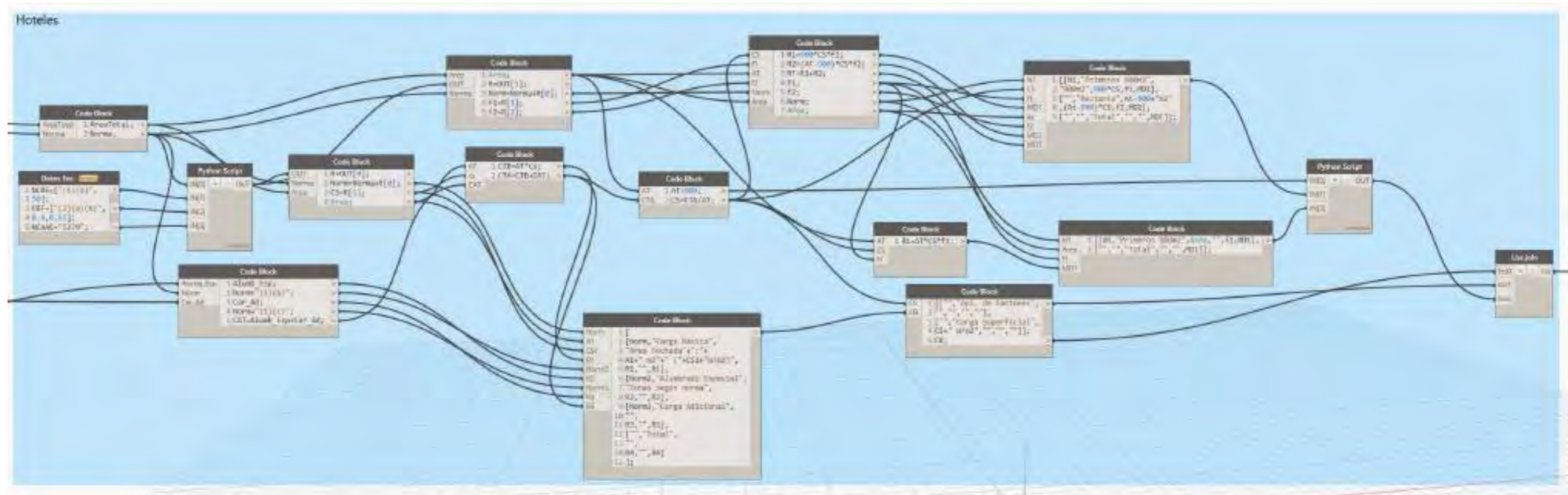
Por otro lado, para el caso de Hoteles, debido al requisito de ingresar cargas adicionales por iluminación especializada aparte de la carga adicional común, se propone un conjunto de nodos aparte de los ya desarrollados para los casos de Escuelas y Hospitales.

Las Figuras 5.47 y 5.48 muestran el proceso de cálculo programado para los casos de Escuelas, Hospitales y Hoteles, incluyendo la implementación del Algoritmo 5.10. El código fuente completo se puede consultar en el Anexo A - Código A.7.



Figura 5.48

Cálculo de la Carga para Hoteles en Dynamo



Fuente: Implementación propia.

Para los Tipos de Edificación no contemplados por la *Sección 050* del CNE-U pero que pueden ser calculados utilizando la Tabla 14 del mismo documento, la aplicación en Dynamo sigue una operación similar a cuando se realizan cálculos con datos de Potencia Superficial. Sin embargo, la consideración de la carga adicional no requiere un tratamiento especial como en los casos de Viviendas Unifamiliares, Departamentos u Hoteles.

En el proceso de programación, además de la interfaz de entrada de datos que permite la selección del Tipo de Edificación, es necesario agregar una base de datos que contenga los valores de potencia superficial y factores de demanda, tanto para alimentadores como para acometidas (Ver Figura 5.49).

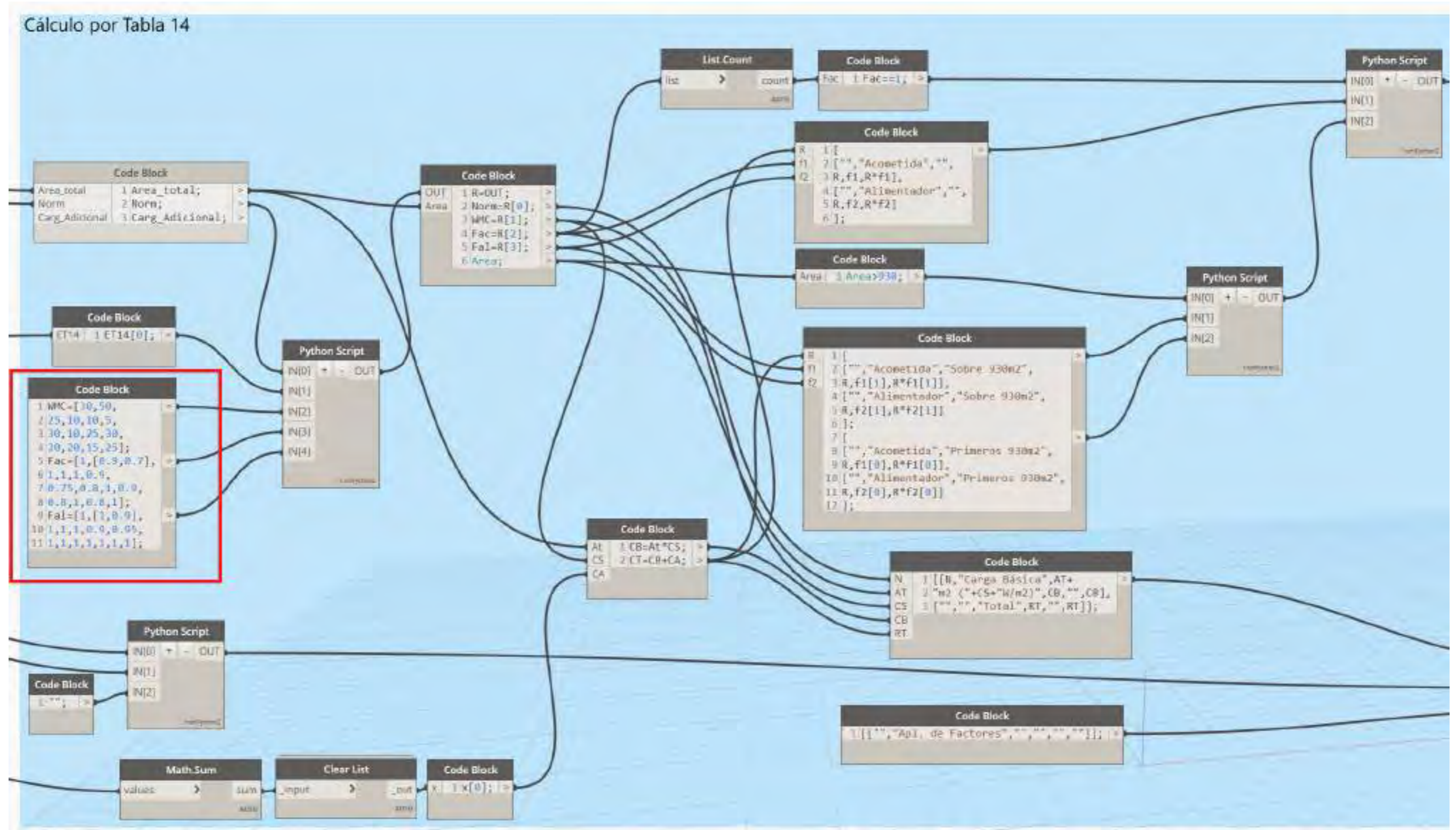
En este apartado, también se debe tener en cuenta que el cálculo de la carga adicional se realiza de manera diferente, y es necesario reorganizar la lógica de programación en función de las cargas reordenadas previamente, que no están consideradas como cargas de uso “doméstico”.

En la Figura 5.50 se muestra una red de nodos que utilizan los datos previamente tomados en la interfaz de selección de cargas adicionales para realizar el cálculo por separado. Se toma en cuenta que, en caso de que no existan o no se seleccionen cargas en la interfaz de Revit, simplemente no se procederá al cálculo.

Por último, mediante sentencias condicionales basadas en los cálculos realizados previamente, se genera la tabla dentro del Software BIM (Ver Figura 5.51).

Figura 5.49

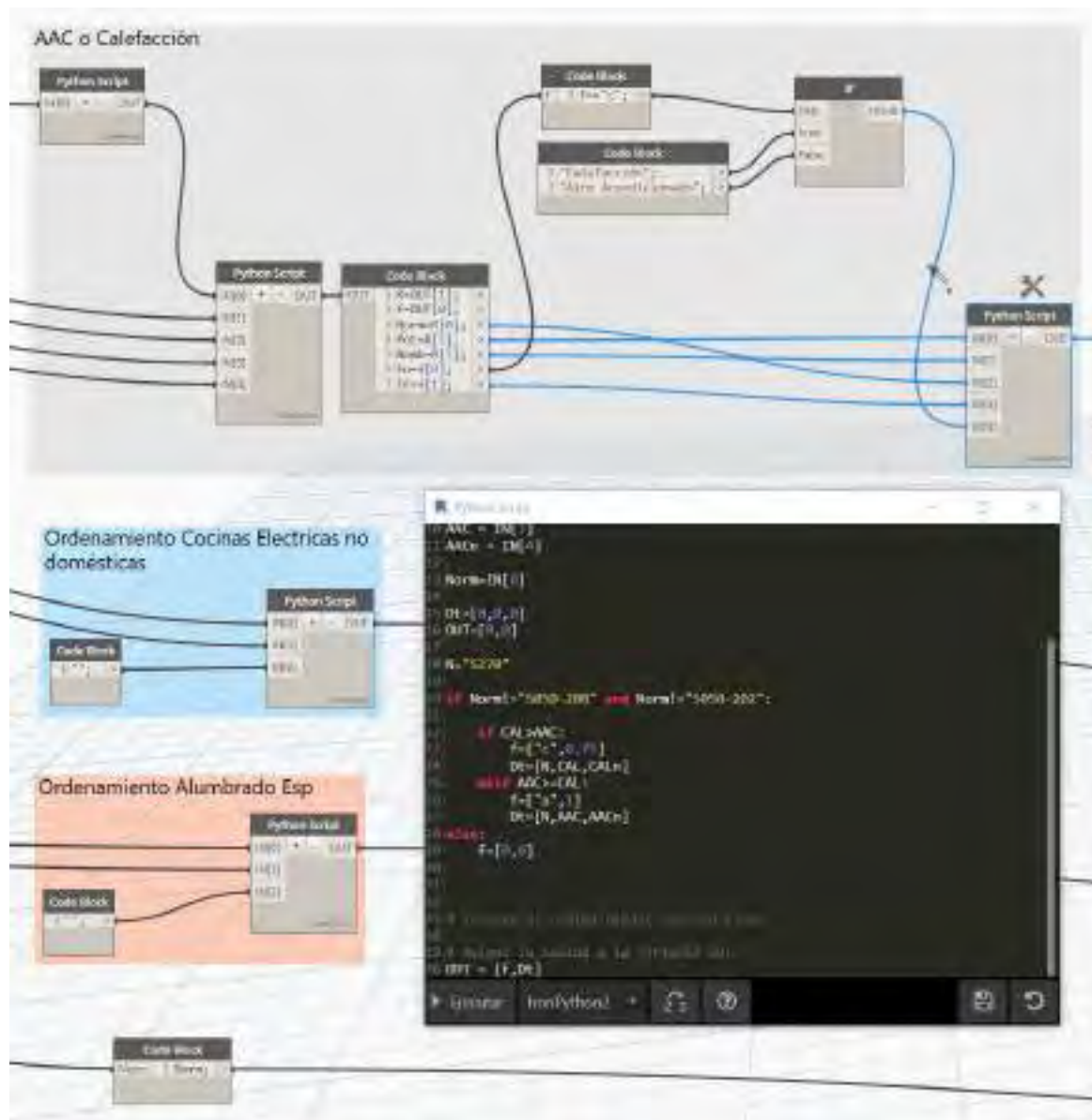
Cálculo de la Carga según Tabla 14 en Dynamo



Fuente: Implementación propia.

Figura 5.50

*Cálculo de Cargas Adicionales de uso NO doméstico en Dynamo*

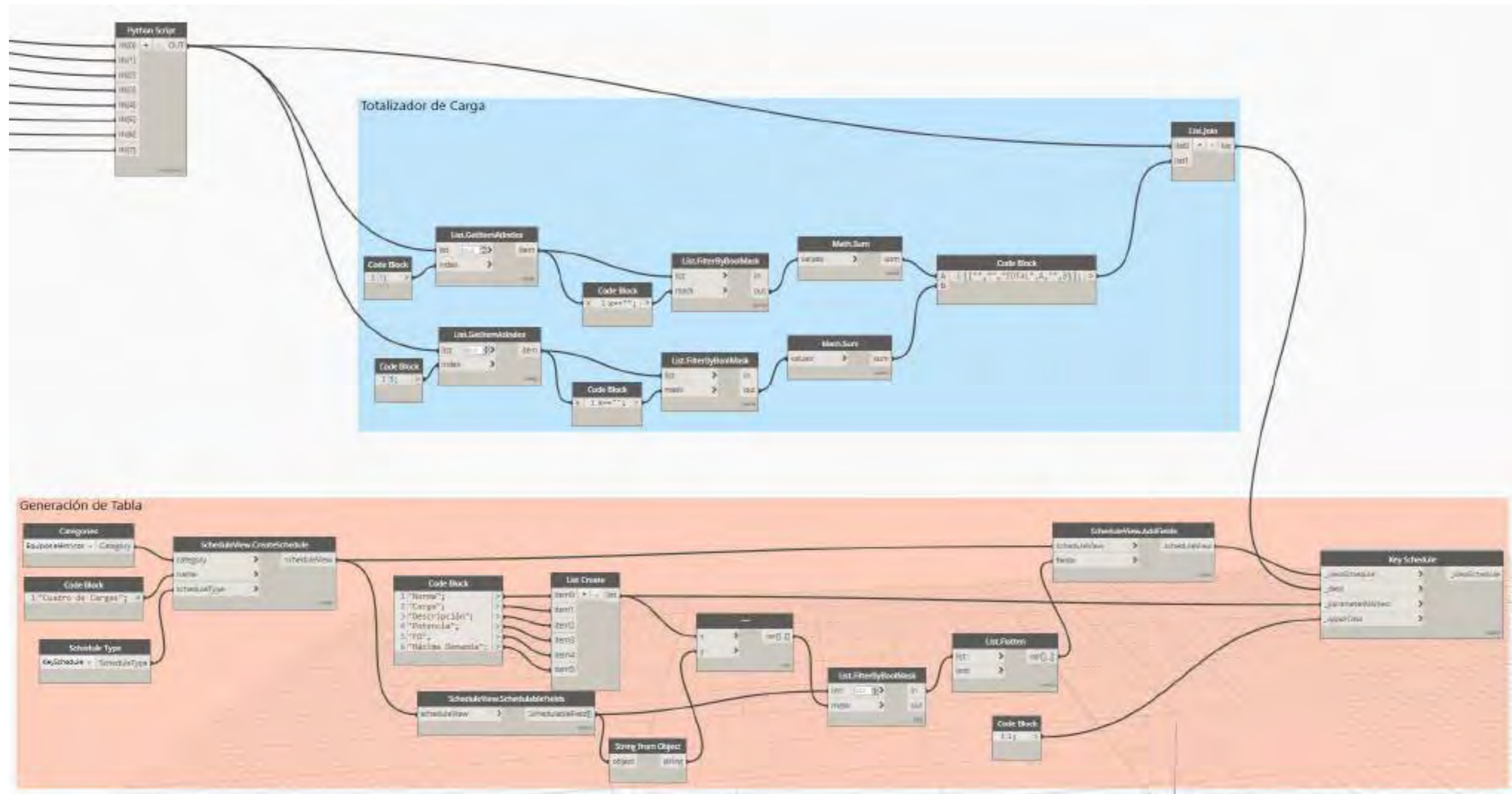


*Fuente:* Implementación propia.



Figura 5.51

Generación del Cuadro de Cargas en Dynamo



Fuente: Implementación propia.

### 5.4.3 Desarrollo del Script “Caídas de Tensión”

Para el desarrollo del script correspondiente al cálculo de caídas de tensión, es necesario tomar en cuenta el diagrama de flujo propuesto en la Figura 5.39 y, al mismo tiempo, implementar el funcionamiento del algoritmo descrito en el Algoritmo 5.12 en el lenguaje de Python, el cual se encuentra detallado en el Anexo A - Código A.8.

Adicionalmente, es necesario utilizar nodos de paquetes desarrollados por terceros, que son los siguientes:

- Paquetes Adicionales
  1. *Orchid* por erfajo
  2. *MEPover* por tpoover
  3. *Spring Nodes* por dimitar.ven
  4. *Genius Loci* por Alban de Chasteigner<sup>10</sup>
  5. *Data-Shapes* por Elayoubi Mostafa
  6. *archi-lab* por Konrad K Sobon

La ejecución del script comienza con una interfaz que permite ingresar datos directos del panel o tablero eléctrico desde el cual se iniciará el cálculo. Además, se requiere una selección ordenada de los conductos eléctricos asociados a los circuitos (ver Figura 5.52).

El ordenamiento de las selecciones es importante debido a que en ciertas situaciones computacionales, *Dynamo* puede reordenar las selecciones de forma aleatoria, lo que afectaría el cálculo de caída de tensión. Dado que cada cálculo de caída de tensión depende del impacto matemático de los parámetros de distancia de los conductos, esta estrategia de ordenamiento manual es la más adecuada.

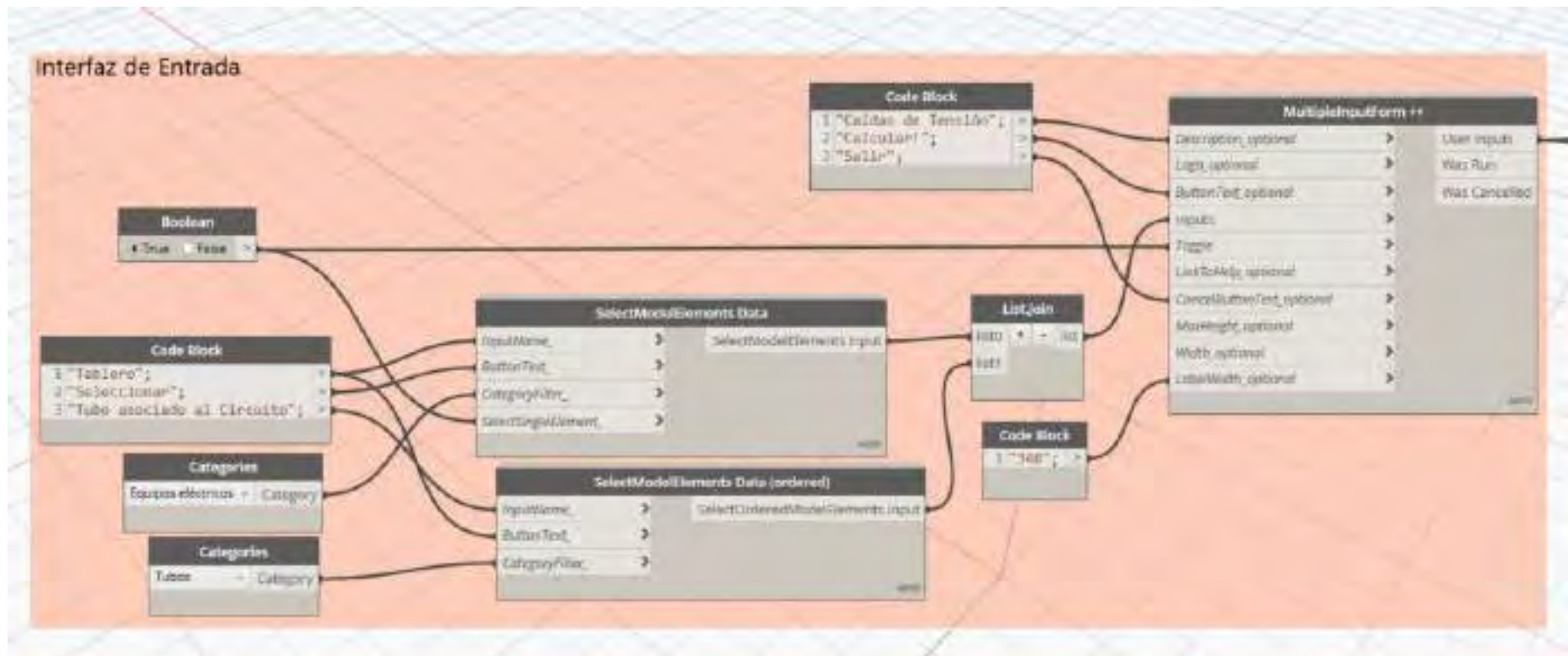
Hasta el momento, Revit no relaciona directamente los circuitos con los conductos correspondientes, por lo que esta operación requiere interacción manual por parte del usuario.

---

<sup>10</sup> Repositorio del autor: <https://github.com/albandechasteigner>

Figura 5.52

Interfaz para la selección de tableros y conductos para el cálculo de caídas de tensión



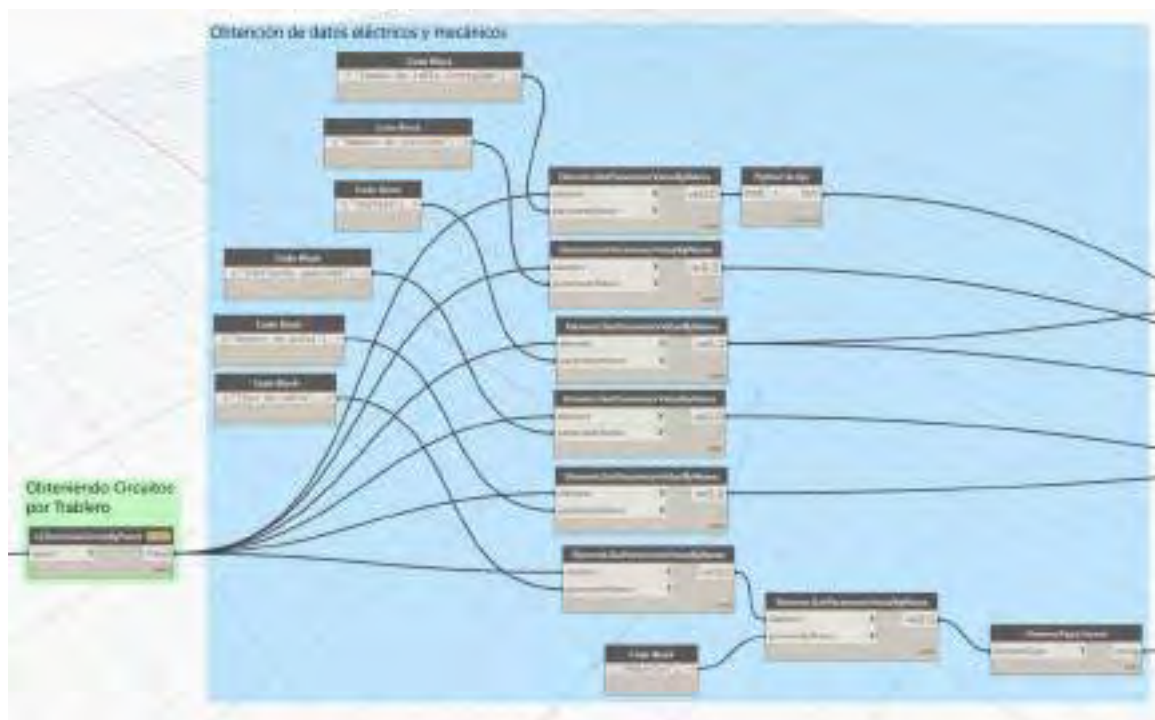
Fuente: Implementación propia.

El diagrama de flujo de Figura 5.39 muestra que, a partir de la toma de datos en la interfaz, el cómputo se bifurca en dos secciones. Una primera sección opera directamente con los parámetros eléctricos de cada circuito perteneciente al tablero seleccionado. La segunda sección se enfoca en obtener el parámetro geométrico necesario para el cálculo, que en este caso es la longitud de los conductos asociados a los circuitos.

Para la primera sección, la Figura 5.53 muestra una vista de *Dynamo* donde se establece la extracción de los datos necesarios para el cálculo. La interfaz de programación visual permite aprovechar el tratamiento de listas con dimensiones similares, lo que facilita el enlace entre los nodos sin necesidad de recurrir a la programación tradicional.

**Figura 5.53**

*Extracción de datos eléctricos de circuitos para el cálculo de caídas de tensión*



*Fuente:* Implementación propia.

En la otra sección, el funcionamiento se basa en la toma de elementos conectados al conducto seleccionado. A partir de ahí, se realiza un filtrado por cada sublista generada, correspondiente a cada circuito asociado al tablero seleccionado en la interfaz. Con los elementos ordenados, es posible extraer la longitud de cada elemento y, de esa forma, calcular las distancias totales (Figura 5.54).

Con los resultados obtenidos en las secciones previas, se procede a implementar el algoritmo de cálculo de caídas de tensión (Algoritmo 5.12), tal como se muestra en la Figura 5.55.

**Figura 5.54**

*Obtención de longitud total por tramos de circuitos eléctricos asociados a sus respectivos conductos*



*Fuente:* Implementación propia.

En la etapa final del cálculo de caídas de tensión, los resultados reciben dos tratamientos. El primero de ellos se muestra en la Figura 5.56, y su objetivo es proporcionar una evaluación rápida y visual del modelo BIM.

En la etapa final, los resultados del cálculo de caídas de tensión se tabulan de forma similar al script “Cargas”, utilizando los datos obtenidos y generando una nueva tabla manipulable para el software BIM, como se muestra en la Figura 5.57.

Figura 5.55

Aplicación del Algoritmo 5.12 para el desarrollo del script “Caídas de Tensión”

The image shows a software interface with two main components: a 'Code Block' and a 'Python Script' window.

**Code Block:**

Variable	Index
Tamaño de Conductor	1
Nombre del Circuito	2
Voltaje	3
Corriente	4
Num de Polos	5
Material del Conductor	6
Longitudes	7

**Python Script:**

```

16 tam=len(Cir)
17
18
19 M=[["Circuito","I","L","Mat","Seccion","DV"] for i in range(tam)]
20
21
22 for i in range(tam):
23     if NP[i]==1:
24         k=1
25     elif NP[i]==3:
26         k=3**(1/2)
27     if Mat[i]=="Cobre":
28         p=0.0176
29     elif Mat[i]=="Aluminio":
30         p=0.0268
31
32
33
34
35 DV=k*I[i]*1.25*(Long[i]/S[i])*8.9*p
36
37
38 M[i][0]=Cir[i]
39 M[i][1]=I[i]*1.25
40 M[i][2]=Long[i]
41 M[i][3]=Mat[i]
42 M[i][4]=S[i]
43 M[i][5]=DV

```

Fuente: Implementación propia.

**Figura 5.56**

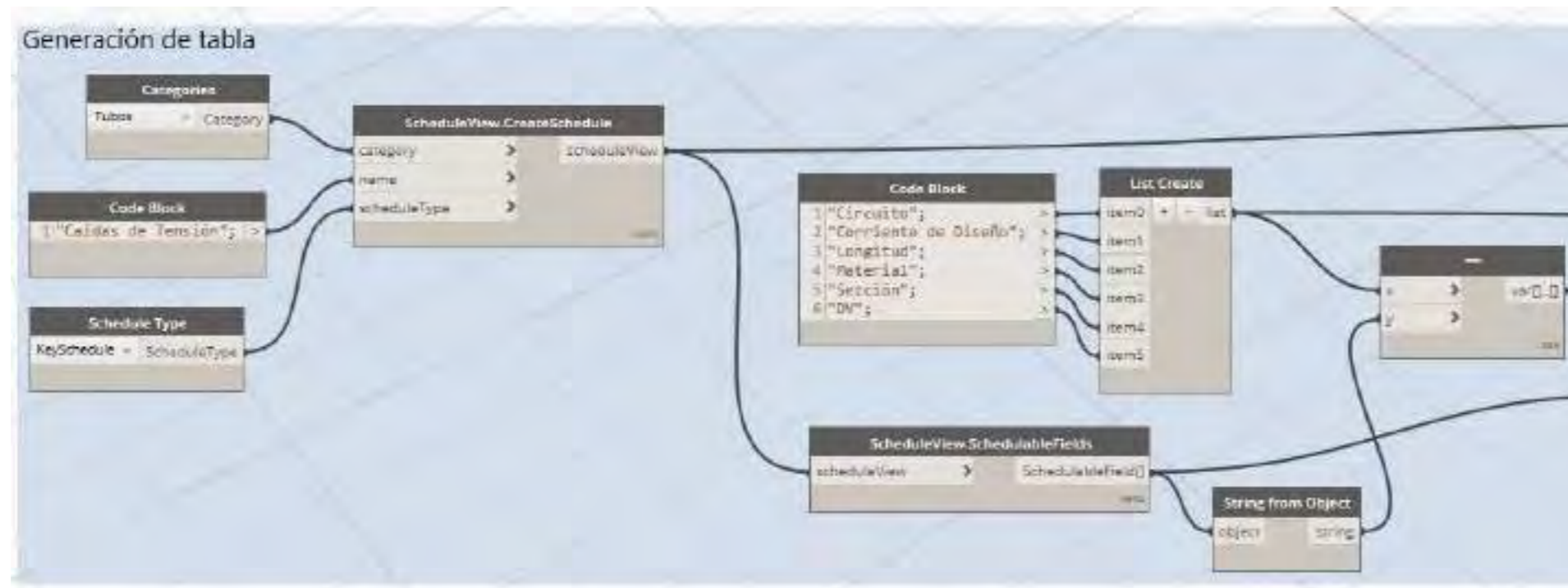
*Generación de rango de colores en función a las caídas de tensión obtenidas para visualización en modelo BIM*



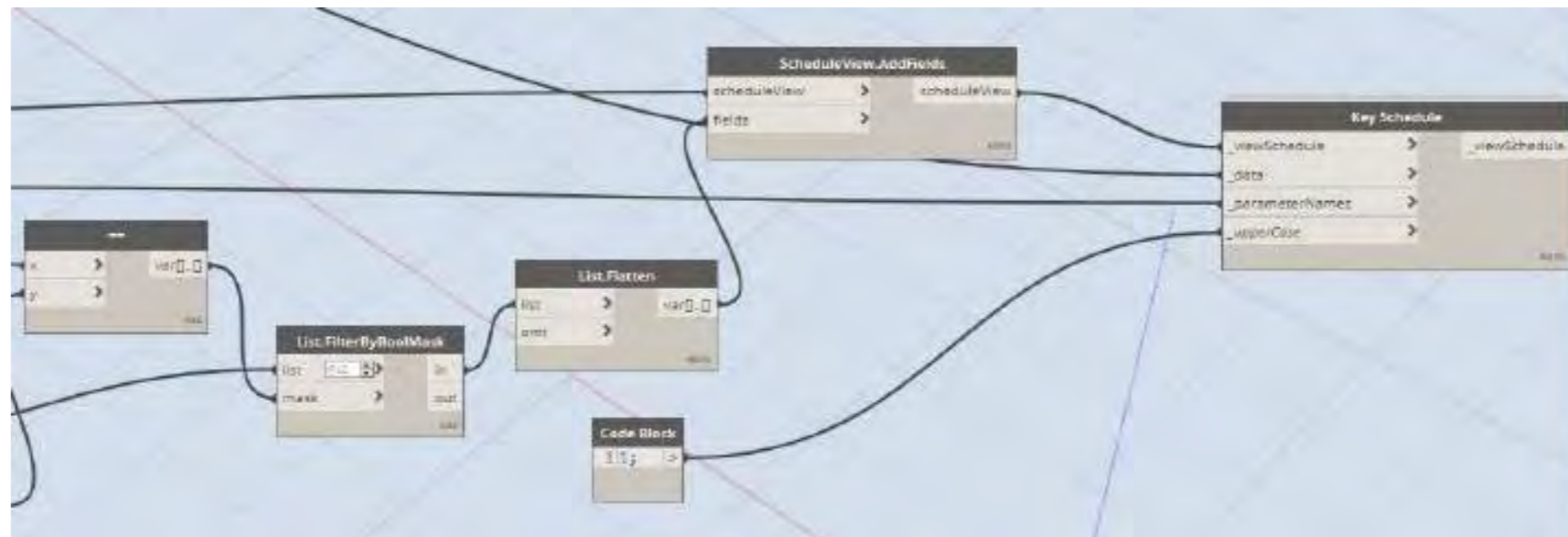
*Fuente:* Implementación propia.

Figura 5.57

Generación del cuadro de caídas de tensión por tablero y circuito



(a) Primera parte del script



(b) Segunda parte del script

Fuente: Implementación propia.



## CAPÍTULO VI

### APLICACIÓN DE LA PROPUESTA

#### 6.1 Introducción

En este capítulo, como parte de los objetivos de la tesis para su aplicación en un contexto académico y profesional, se explica el funcionamiento y los resultados de los scripts desarrollados en el capítulo anterior. Se enfoca en obtener vistas con la calidad técnica de un modelo BIM y abarcar el trabajo de ingeniería relacionado con los cálculos específicos para el diseño de instalaciones eléctricas de interiores.

#### 6.2 Contexto Aplicativo

En el contexto aplicativo de la propuesta, se destaca la importancia del software BIM *Autodesk Revit 2022*, donde se ejecutan los scripts desarrollados. Es relevante mencionar que el objetivo no es reemplazar las funciones del software, sino complementar y potenciar el proceso de diseño y cálculo mediante el uso de herramientas adicionales.

Se abarcan características BIM, como el estudio de necesidades, la incorporación de información relevante y la posibilidad de gestionar el proyecto de manera más efectiva. Sin embargo, es fundamental comprender que el alcance de la aplicación se limita a la tercera dimensión (visualización) dentro del contexto BIM, ya que los scripts se han programado con objetivos específicos en esta dimensión.

#### 6.3 Base Arquitectónica

En el saber de que un proyecto de instalaciones eléctricas interiores requiere como base fundamental el diseño arquitectónico desarrollado por un profesional, para la aplicación de la propuesta se posee la siguiente base arquitectónica.

- *Proyecto* : Edificio Comercial–Departamental “Unión”
- *Profesional responsable* : Arq. Silvia Luciana Jordan Ramos.
- *CAP* : 014989

El documento el cual autoriza el uso de esta base arquitectónica se puede consultar en el Anexo C.

### 6.3.1 *Características*

La base arquitectónica sobre la cual se pretende dar aplicación a la propuesta desarrollada en la presente tesis, posee las siguientes características:

- *Uso* : Comercial y residencial
- *Área del Terreno* : 144.75 m<sup>2</sup>
- *Área del Techada Total* : 698.00 m<sup>2</sup>
- *Niveles* : 6 Niveles y 1 azotea
- *Descripción* : 2 Niveles para uso comercial  
4 Niveles para uso residencial
- *Otras Áreas* : 1 Cuarto de Bombeo  
1 Cuarto de Máquinas

## 6.4 Ejecución de Módulos

### 6.4.1 *Consideraciones*

Las consideraciones para el uso de la base arquitectónica en la aplicación de la propuesta son las siguientes:

1. La base arquitectónica debe contar con un nivel mínimo de información BIM, que incluya elementos como la estructura, muros y mobiliario. Esta información es necesaria para realizar el diseño de las instalaciones eléctricas desde el punto de vista del estudio de necesidades.
2. La relación entre las disciplinas dentro del proyecto se enfoca principalmente en la interacción entre Arquitectura e Instalaciones Eléctricas. Dado que el alcance BIM de la propuesta se limita a la dimensión 3D, se prioriza la colaboración entre estas dos disciplinas para el diseño de las instalaciones eléctricas.

La Figura 6.1 ilustra la relación entre la disciplina de Arquitectura y las Instalaciones Eléctricas en el contexto de la aplicación de la propuesta.

**Figura 6.1**

*Base Arquitectónica BIM 3D para modelado de instalaciones eléctricas*



*Fuente:* Captura de software propia.

Al utilizar un mismo archivo para trabajar con el modelo BIM, se aprovecha la ventaja de tener todos los alcances arquitectónicos disponibles tal como fueron concebidos originalmente. Esto facilita la detección rápida de incompatibilidades y permite una mayor integración entre la disciplina de Arquitectura y las Instalaciones Eléctricas.

Aunque trabajar en un mismo archivo puede aumentar el peso informático considerablemente, la ventaja de tener toda la información interconectada supera esta limitación. Además, al mantener todo en un solo archivo, se evita la necesidad de realizar intercambios frecuentes de información entre distintos modelos, lo que puede reducir errores y agilizar el proceso de diseño.

El uso de un archivo vinculado en el contexto de la propuesta permite una colaboración más estrecha y una mejor visualización de la relación entre la arquitectura y las instalaciones eléctricas, lo que conduce a un diseño más preciso y coordinado.

#### **6.4.2 *Flujo de Trabajo Propuesto***

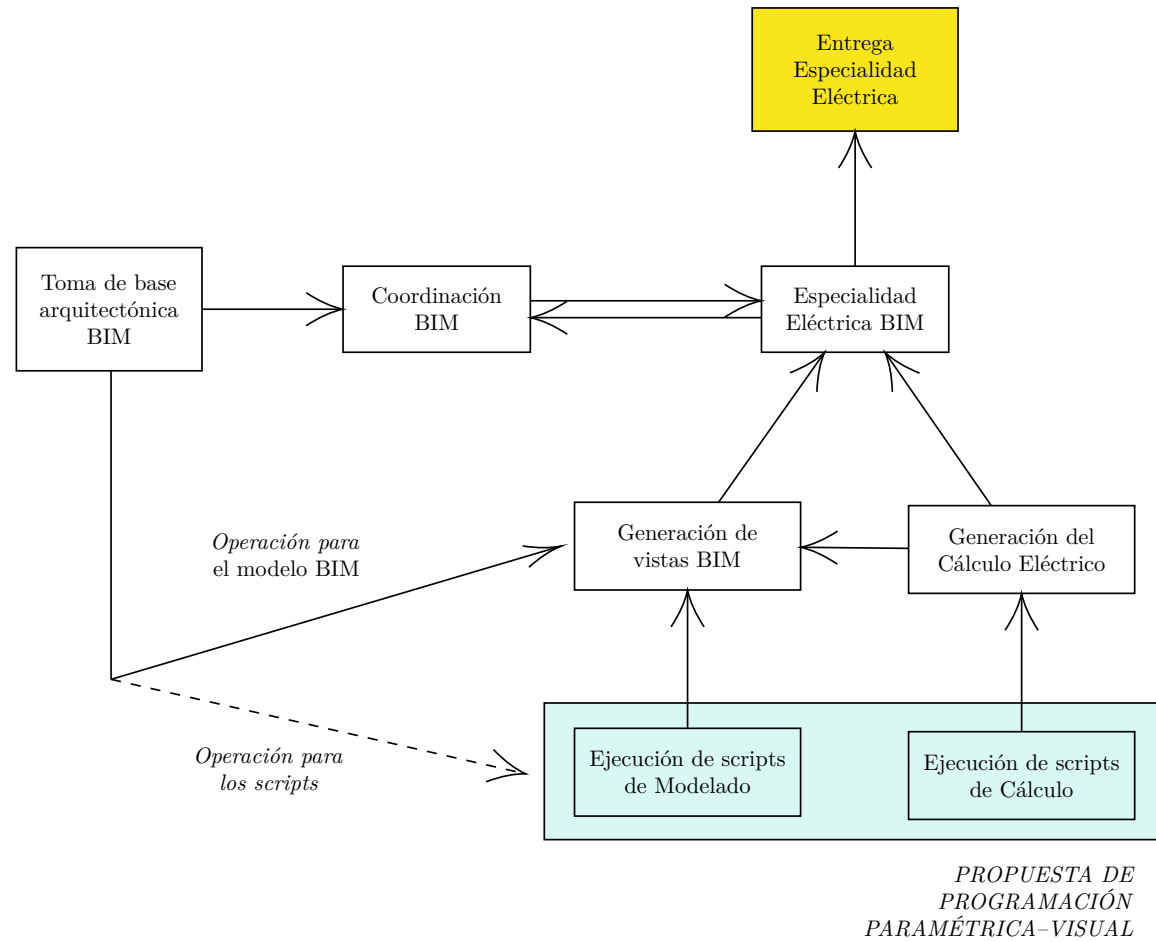
Para el planteamiento de un Workflow (Wf) o Flujo de Trabajo BIM, inicialmente entra a tallar la necesidad propia de ya contar con un modelo BIM de la especialidad de Arquitectura y sobre el cual empezar el diseño asistido por los scripts desarrollados.

Adicionalmente, se entiende también que el presente Workflow tiene base sustento propuesto por Chen et al. (2016), mismo que fue estudiado en el tercer capítulo de la presente tesis. Sin embargo, en el escenario presente donde se pretende la inclusión de herramientas programadas y los entregables que solicita la Norma EM 010 del Reglamento Nacional de Edificaciones (Ministerio de Vivienda, Construcción y Saneamiento, 2006), se propone el flujo de trabajo expuesto en la Figura 6.2.

En ese sentido, se propone un proceso de ejecución en el que se detalla la interrelación entre los scripts desarrollados y su continuidad hacia la obtención de modelos BIM y a su vez a la etapa de diseño y cálculo de instalaciones eléctricas interiores.

**Figura 6.2**

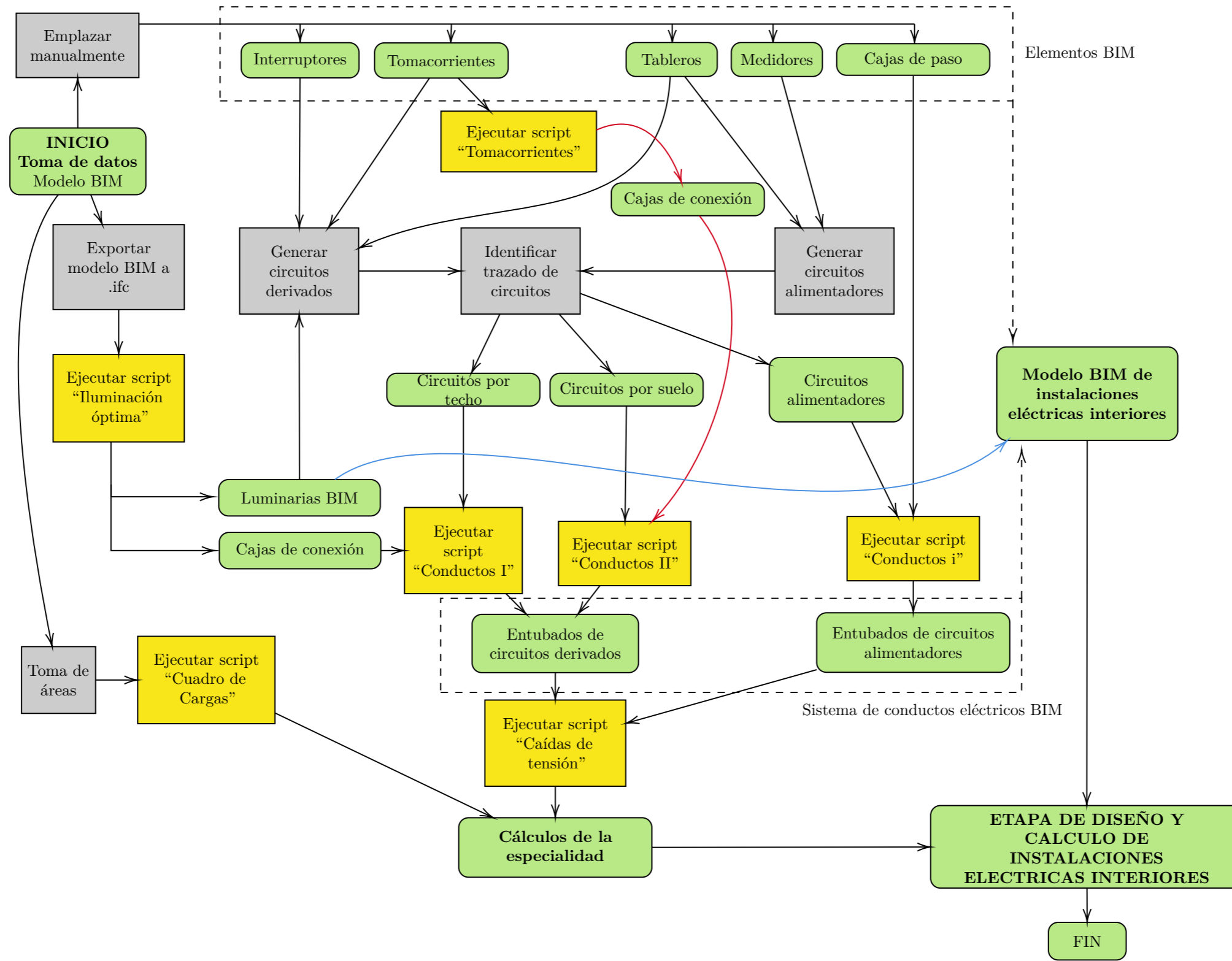
*Workflow BIM para la especialidad de instalaciones eléctricas asistida por programación paramétrica visual, basado en el Workflow de Chen et al. (2016)*



*Fuente:* Elaboración propia.

Figura 6.3

Diagrama de flujo para la ejecución de scripts



Fuente: Elaboración propia.

### 6.4.3 *Planteado de Circuitos*

Esta primera etapa de aplicación implica el uso directo del software BIM *Autodesk Revit 2022* y la explotación de sus propias herramientas.

En este caso, la circuitería y el cableado de los artefactos dependen de un emplazamiento inicial, lo que requiere el desarrollo de esta tarea como parte del flujo de trabajo. En este escenario, se busca incluir el criterio propio del ingeniero electricista en el diseño de las instalaciones eléctricas.

La propuesta consiste en:

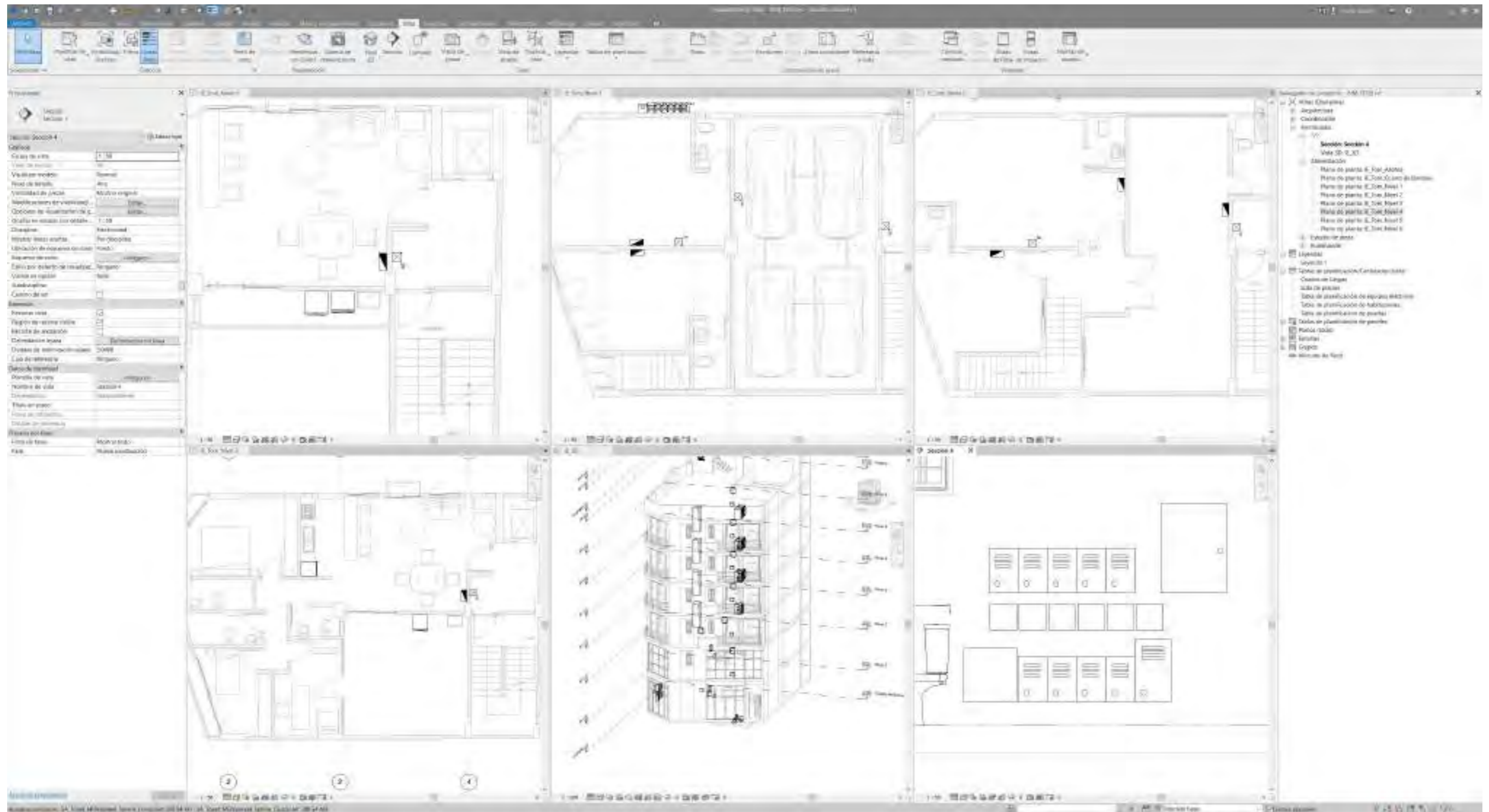
1. Emplazar manualmente las unidades de medidores de energía y tableros de distribución.
2. Establecer ubicaciones para cajas de paso y tomacorrientes de forma manual.
3. Exportar el modelo BIM a un formato *.ifc* para la ejecución del script “Iluminación Óptima”
4. Emplazar manualmente interruptores.
5. Generar circuitería eléctrica por medio de herramientas propias del software BIM.

Dentro del proceso descrito anteriormente, se busca minimizar los tiempos en los cuales se requiera un trabajo técnico de manejo de elementos BIM, mientras que en los procesos donde es necesaria la interacción manual o tradicional dentro del enfoque BIM, sea el criterio profesional el que permita el desarrollo del diseño de instalaciones eléctricas de forma responsable.

Las figuras 6.4 y 6.4 muestran, en primer lugar, la operación manual del BIM, la cual permite incluso tener vistas de detalle que en CAD requería hacer un dibujo por separado. A su vez, el manejo tridimensional del BIM y sus características permiten mover un modelo entre el software BIM y otro de especialidad, como en este caso, el Dialux EVO, que posibilita la aplicación de la propuesta.

Figura 6.4

Emplazamiento Manual de Medidores, Cajas de Paso y Tableros Eléctricos



Fuente: Captura de software propia.



**Figura 6.5**

*Migración de modelo BIM a Dialux por medio de un formato .ifc*



*Fuente:* Captura de software propia.

Aunque el formato *.ifc* dentro del software de Dialux puede presentar alguna imprecisión en las representaciones de líneas de modelo, mantiene las características geométricas de los locales generados. Por tanto, se puede comenzar con el diseño lumínico automatizado o personalizado, teniendo en cuenta la inclusión de la base normativa peruana, que en este caso es el alcance sobre los niveles mínimos de iluminación indicados en la norma *EM 010* del Reglamento Nacional de Edificaciones (Ministerio de Vivienda, Construcción y Saneamiento, 2006).

En la Figura 6.6 se observa la interacción con la norma peruana previamente configurada dentro de Dialux, teniendo además una precisión geométrica dentro del modelo tridimensional que ha sido provisto por el formato *.ifc*, gracias a los potenciales multiformato que poseen los softwares BIM.

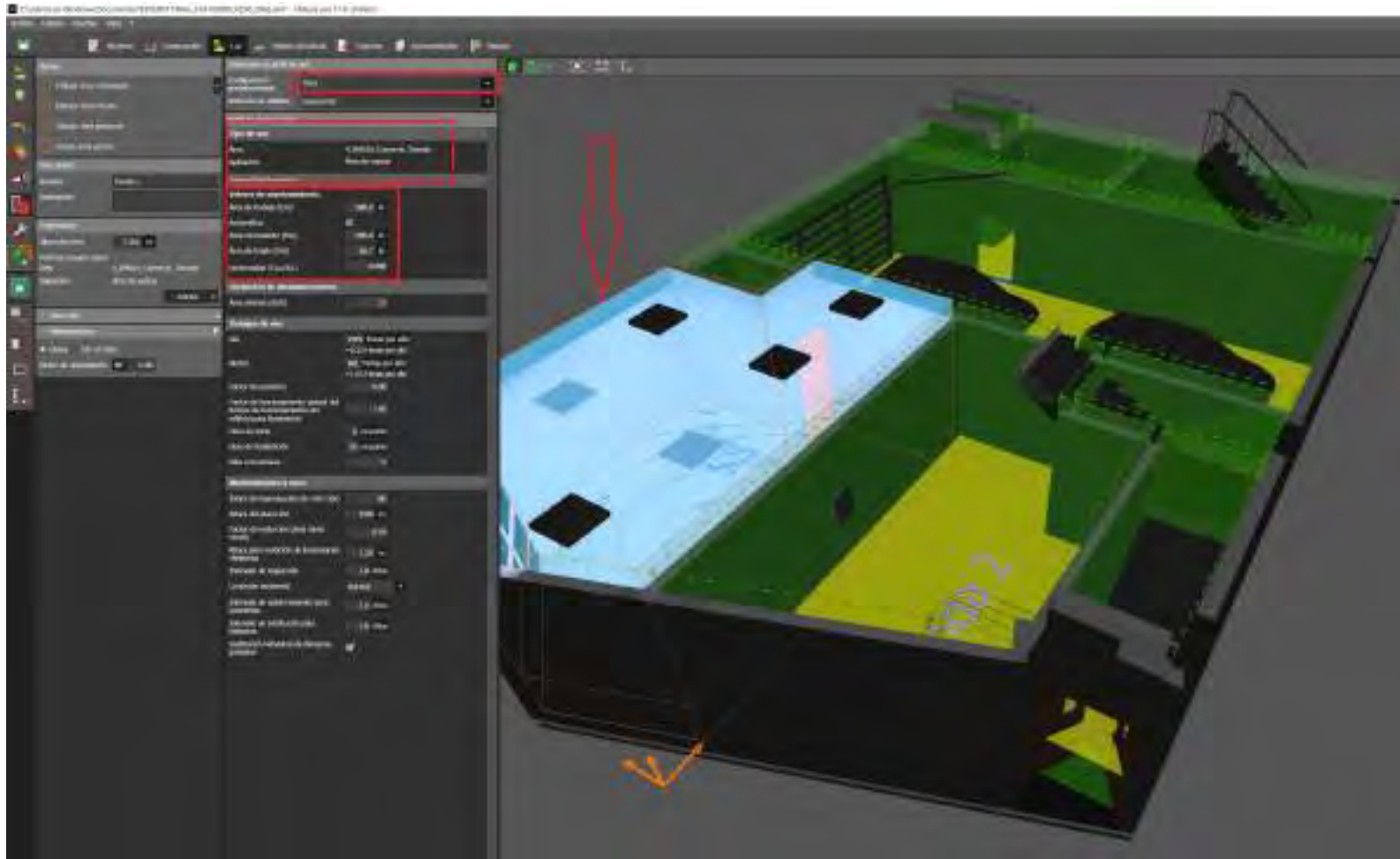
El objetivo de tener esta capacidad de cálculo es optimizar el proceso de diseño, teniendo en cuenta el *Code Checking* que requiere el BIM como metodología.

Ahora bien, el script desarrollado en el capítulo anterior remite su funcionamiento a la operación necesaria con formatos CAD, por lo que es necesaria la exportación de resultados a AutoCAD para que, dentro de la capacidad de este para procesar coordenadas *XYZ*, se puedan obtener los archivos necesarios para la operación entre Dynamo y Revit (Figura 6.7).

Al ejecutar el script de Tharwat (2010) dentro de los resultados de AutoCAD, es posible extraer la información geométrica posicional que permitirá el correcto funcionamiento del script desarrollado para el Software BIM.

**Figura 6.6**

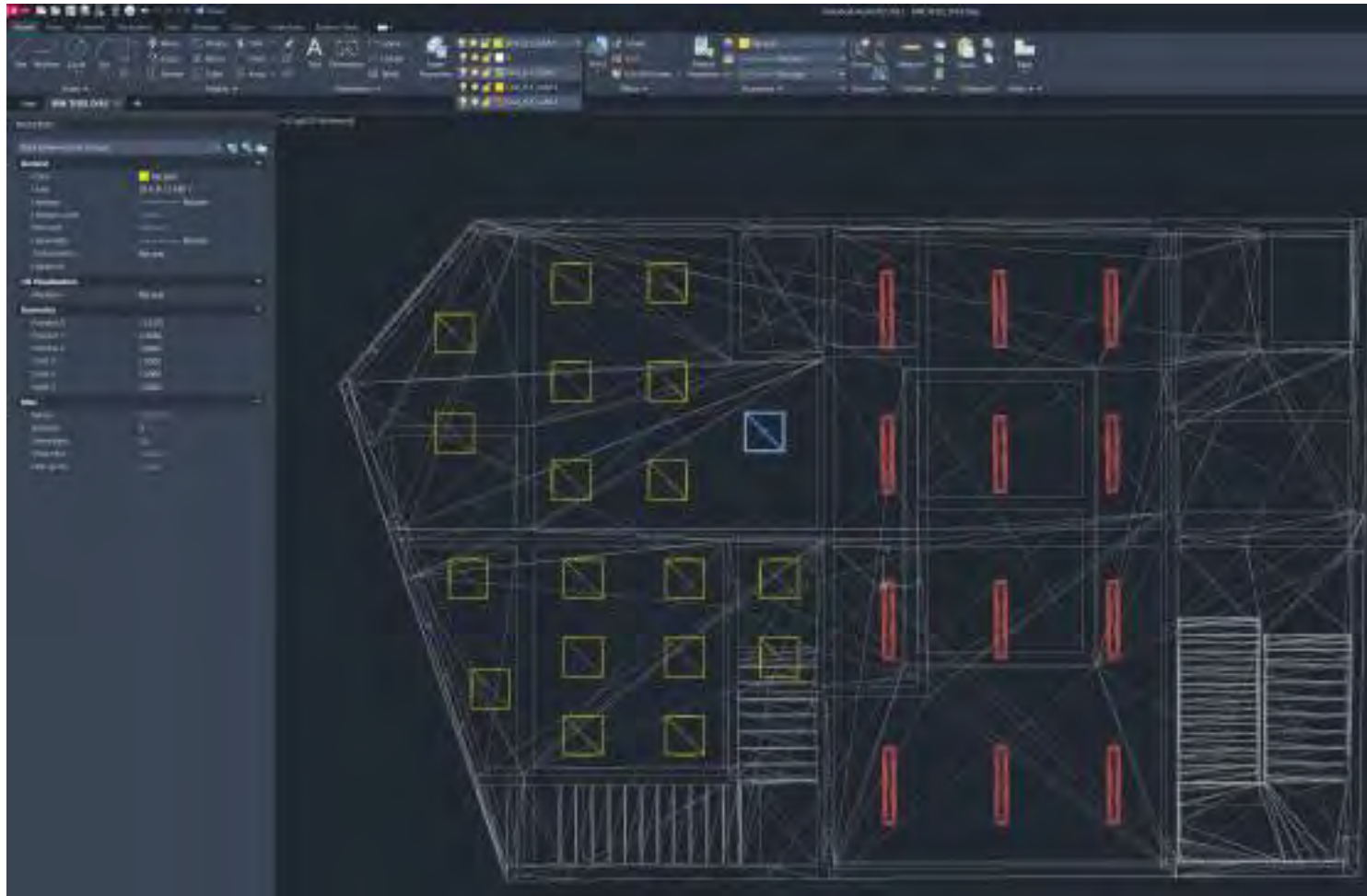
*Diseño lumínico BIM asistido por Dialux con base de formato .ifc*



*Fuente:* Captura de software propia.

**Figura 6.7**

*Posiciones provistas por Dialux para el procesamiento en AutoCAD*



*Fuente:* Captura de software propia.

Figura 6.8

Coordenadas de luminarias extraídas de Dialux por medio del script de Tharwat (2010)

```

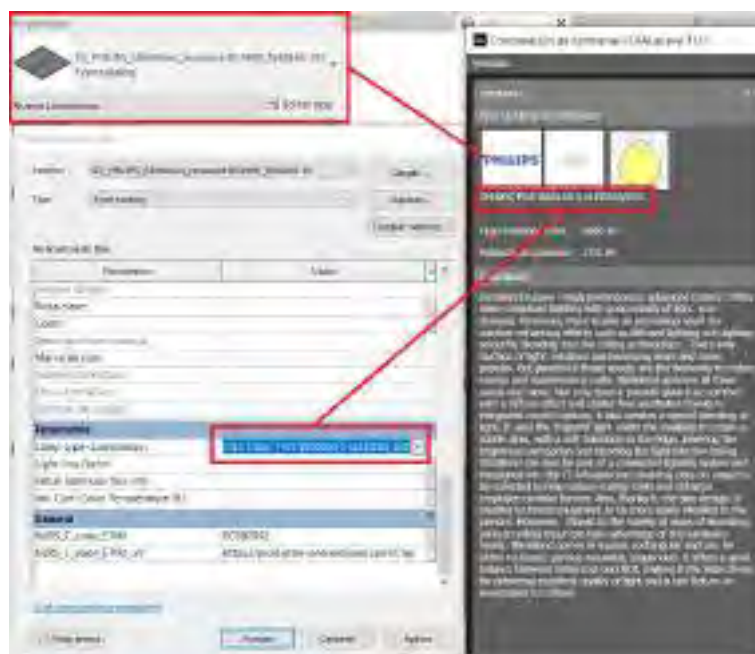
Rinova Soliman Solimato Mr Ayadi
-4.7607,4.6369,3.0000,DLX_FLE_LUM 1
-4.7607,4.6369,3.0000,DLX_FLE_LUM 1
-3.2891,4.6369,3.0000,DLX_FLE_LUM 1
-3.2891,4.6369,3.0000,DLX_FLE_LUM 1
-3.2891,4.6369,3.0000,DLX_FLE_LUM 1
-3.2891,4.6369,3.0000,DLX_FLE_LUM 1
-4.5833,-2.2726,3.3000,DLX_FLE_LUM 2
-4.5833,-2.2726,3.3000,DLX_FLE_LUM 2
-4.5833,-2.2726,3.3000,DLX_FLE_LUM 2
-4.5833,-2.2726,3.3000,DLX_FLE_LUM 2
-3.0302,-2.2726,3.3000,DLX_FLE_LUM 2
-3.0302,-2.2726,3.3000,DLX_FLE_LUM 2
  
```

Fuente: Captura de software propia.

Es importante destacar que las características de las luminarias y lámparas utilizadas en Dialux deben ser coherentes con las utilizadas en el modelo BIM. En este sentido, se aprovechan las librerías BIM desarrolladas por los fabricantes, que proporcionan datos precisos y actualizados de los productos, como se observa en la Figura 6.9.

Figura 6.9

Igualdad de luminarias BIM en Dialux y Revit



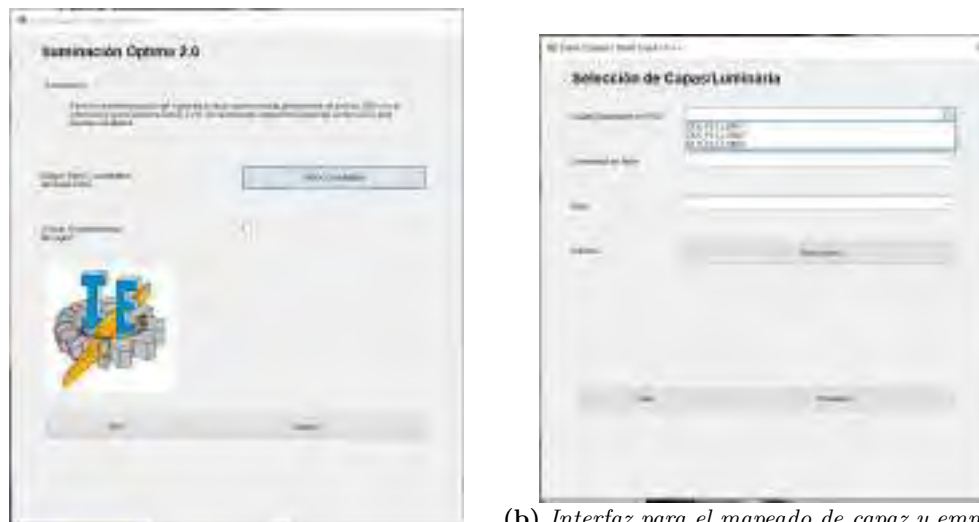
Fuente: Captura de software propia.

La ejecución del script se lleva a cabo a través de una interfaz visual previamente desarrollada en los bloques y enlaces de *Dynamo*, como se muestra en la Figura 6.10. Esta interfaz permite operar con un ambiente o “local” a la vez, lo que significa que se pueden realizar cálculos y diseños lumínicos

de forma individual para cada espacio dentro del modelo BIM. De esta manera, se logra una mayor precisión y personalización en el diseño de iluminación para cada área específica del proyecto.

### Figura 6.10

*Interfaz de ejecución del script "Iluminación Óptima"*



(a) *Interfaz de entrada de datos de coordenadas mientro en Revit de luminarias provistas por Dialux y AutoCAD*

(b) *Interfaz para el mapeado de capas y emplazamiento en Revit*

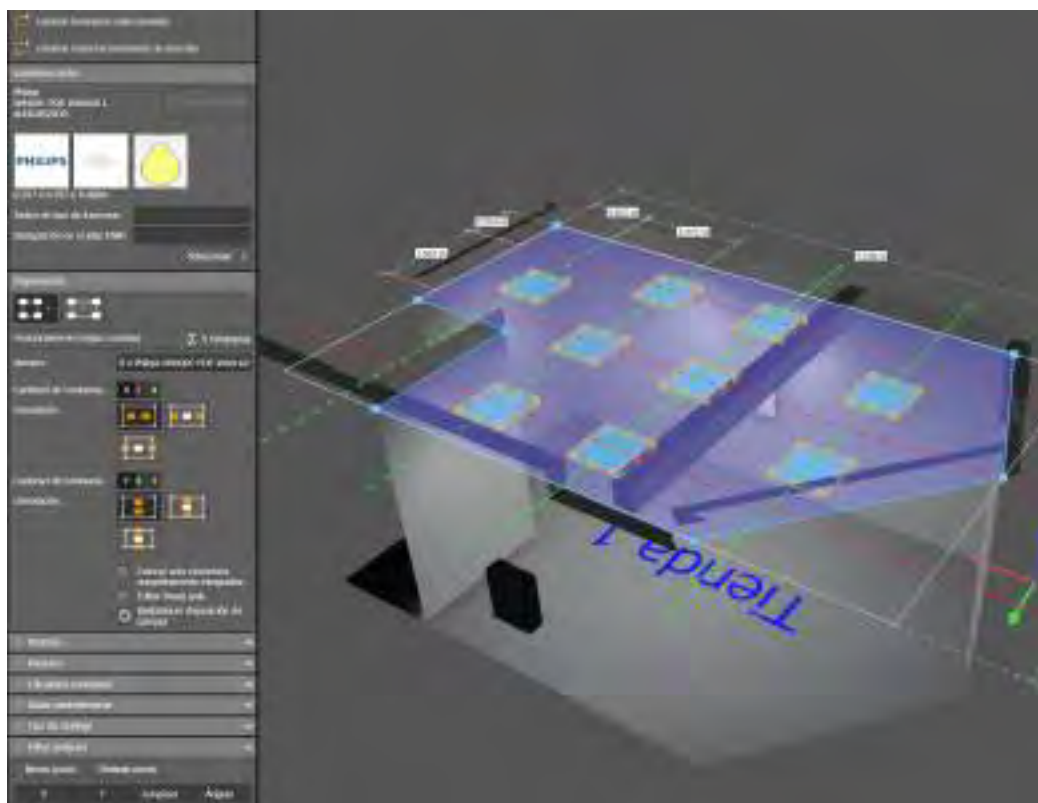
*Fuente:* Captura de software propia.

En la Figura 6.11 se presenta una comparativa entre las posiciones geométricas calculadas y diseñadas en Dialux (Figura 6.11a) y una vista BIM generada gracias a la ejecución de la propuesta (Figura 6.11b). Esta operación puede ser aplicada a los demás ambientes donde se haya realizado un diseño lumínico personalizado para asegurar el cumplimiento preciso de la normativa establecida.

La capacidad de realizar esta comparación entre los modelos de Dialux y el modelo BIM resulta valiosa para verificar y validar la precisión de los cálculos lumínicos y asegurar que se cumplan los niveles mínimos de iluminación requeridos por la norma. Además, la integración de los resultados de Dialux en el modelo BIM permite tener una visión más completa y detallada del proyecto, facilitando la toma de decisiones y optimizando el proceso de diseño.

**Figura 6.11**

*Comparativa BIM tras la ejecución del script "Iluminación Óptima"*



(a) *Diseño lumínico desarrollado en Dialux*



(b) *Emplazamiento BIM automatizado en Revit*

*Fuente:* Captura de software propia.

Adicionalmente, si se recurre a la interfaz de *Dynamo* (Figura 6.12) para conocer los tiempos se ejecución se obtuvo un resultado de:

$$t_{ejecución} = 22\,664\text{ ms}$$

A través de la misma interfaz (Figura 6.13) se puede consultar los tiempos por nodos y de tal modo, se tiene el siguiente desglose:

$$t_{Interfaz1} = 6960\text{ ms}$$

$$t_{Interfaz2} = 13\,024\text{ ms}$$

Por lo tanto, considerando que el tiempo que se toma en la interfaz es debido a la operación del usuario, se puede obtener el tiempo neto de ejecución del programa como:

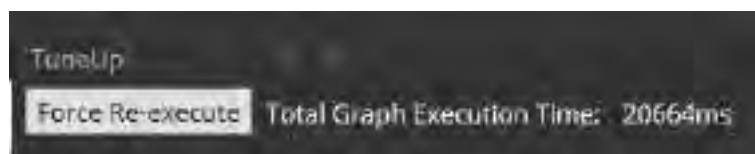
$$t_{Neto} = t_{ejecución} - (t_{Interfaz1} + t_{Interfaz2})$$

Consecuentemente, el tiempo neto para la ejecución del script en un ambiente de 26,84 m<sup>2</sup>, con un emplazamiento de nueve (9) unidades de iluminación, se obtiene un tiempo neto de cómputo de:

$$t_{Neto} = 2680\text{ ms}$$

### Figura 6.12

*Tiempo de ejecución del script "Iluminación Óptima"*



*Fuente:* Captura de software propia.



Figura 6.13

Tiempo de ejecución por nodo del script “Iluminación Óptima”

TimeLog

Forced Re-execute Total Graph Execution Time: 20664ms

#	Name	Execution Time (ms)
44	Logic Block	1
50	String Split	1
33	if	1
99	List Filter	1
72	List.GetItemAtIndex	1
83	List.FilterByBoolMask	1
91	PrintByColorName	1
94	String To Number	1
35	Code Block	1
85	List.GetItemAtIndex	1
41	Categorías	1
81	Code Block	1
82	List.GetItemAtIndex	1
60	Code Block	1
99	Code Block	1
28	Code Block	1
98	Code Block	1
77	List.GetItemAtIndex	1
74	Code Block	1
84	List Filter By Bool Mask	2
66	Code Block	2
63	DropDown Data	3
96	All Family Instance By Face And Point	5
37	Layers Get All Level	6
25	Boolean Input	7
29	Texting Data	7
10	of Family By Category	7
38	DropDown Data	8
80	List Filter By Bool Mask	8
33	Drop Down Data	8
58	List by properties	10
8	of Family By Category	13
27	File Path Data	16
12	DropDown Data	18
38	Send Actual Elements Data	25
76	Level By Name	60
92	All Family Instance By Face And Point	86
42	All Family Types Of Category	92
73	Element Faces	115
15	Element Properties	120
71	Multiple Input Form ++	6500
67	Multiple Input Form ++	13000
Execution Time On Current Run		20664

Fuente: Captura de software propia.

Luego del proceso de diseño lumínico en Dialux y la obtención de resultados precisos y compatibles mediante la integración con el script desarrollado, se procede a trasladar la información al entorno de *Revit*. En este entorno, específicamente en el ambiente *Electrical*, se puede llevar a cabo un diseño de circuitería automatizado.

Una vez ubicado en el ambiente *Electrical* de *Revit*, es posible dotar de potencia eléctrica al panel correspondiente de diseño, como se muestra en la Figura 6.14. De esta manera, el proceso de diseño se puede propagar a las plantas siguientes, tomando en cuenta el mejor planteamiento de circuito realizado por *Revit*.

Este proceso de diseño automatizado y la propagación de circuitos a través de las plantas siguientes permite tener un planteamiento de entubado más preciso.

De igual manera, se procede con los circuitos restantes, aprovechando el soporte que proporciona la metodología BIM. Como se puede apreciar en la Figura 6.15, se tiene un control total sobre las características eléctricas que se están considerando en el proceso de diseño del proyecto.

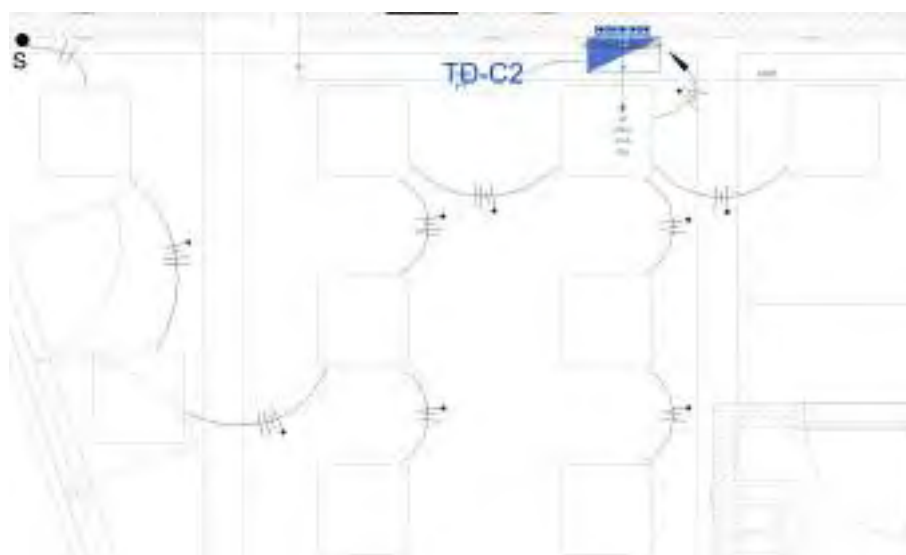
Esta capacidad de control y manejo de datos eléctricos dentro del modelo BIM será de gran ayuda en etapas posteriores del proyecto. Permitirá complementar los cálculos y automatizar la selección de materiales, como dispositivos de seguridad y calibres de conductores, ya que se cuenta con información relevante, como la potencia conectada y la corriente aproximada, disponibles en la biblioteca BIM del proyecto.

**Figura 6.14**

*Generación del circuito eléctrico por medio de herramientas BIM.*



(a) *Suministro de potencia eléctrica a un circuito por medio del entorno Electrical de Revit*

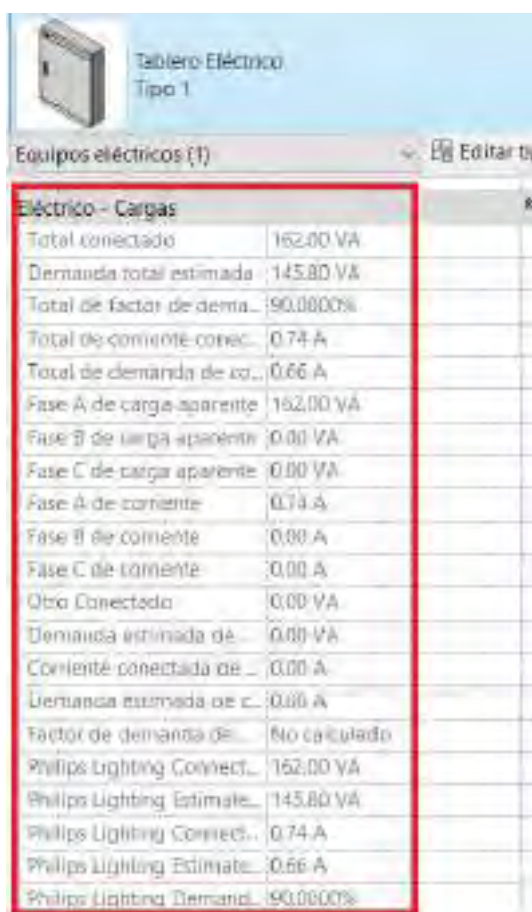


(b) *Generación del circuito de iluminación automático junto a una propuesta de cableado.*

*Fuente: Captura de software propia.*

Figura 6.15

Datos BIM de tipo eléctrico-cargas generados para un tablero eléctrico.



Eléctrico - Cargas	
Total conectado	162.00 VA
Demanda total estimada	145.00 VA
Total de factor de dema.	90.0000%
Total de corriente conect.	0.74 A
Total de demanda de cor.	0.66 A
Fase A de carga aparente	162.00 VA
Fase B de carga aparente	0.00 VA
Fase C de carga aparente	0.00 VA
Fase A de corriente	0.74 A
Fase B de corriente	0.00 A
Fase C de corriente	0.00 A
Otro Conectado	0.00 VA
Demanda estimada de ...	0.00 VA
Corriente conectada de ...	0.00 A
Demanda estimada de c.	0.00 A
Factor de demanda de ...	No calculado
Philips Lighting Connect.	162.00 VA
Philips Lighting Estimate.	145.00 VA
Philips Lighting Connect.	0.74 A
Philips Lighting Estimate.	0.66 A
Philips Lighting Demand.	90.0000%

Fuente: Captura de software propia.

#### 6.4.4 Visualización de Conductos

Dentro del proceso BIM planteado en el Workflow (Wf) de la Figura 6.2, el trabajo implica la obtención completa de vistas BIM, así como el cálculo basado en el diseño. Por lo tanto, además del emplazamiento del equipo eléctrico a conectar dentro de la instalación eléctrica, es necesario generar vistas relacionadas con los conductos requeridos para la instalación.

Dado que se utilizaron herramientas propias del software BIM para el planteamiento automático de circuitos, el modelado de conductos debe realizarse a partir de este trabajo previo. La representación gráfica de un circuito proporciona una interpretación de la instalación de conductos en función de la propuesta circuital previa.

En este contexto, se propone el siguiente proceso a aplicar en situaciones de diseño que requieran el modelado de conductos BIM:

1. Identificar el circuito a generar respectivo entubado.
2. Identificar los puntos donde se requieran cajas de conexión específicas de uso, por ejemplo, cajas de paso en montantes de alimentación eléctrica, cajas octogonales en circuitos de iluminación o cajas rectangulares para circuitos de tomacorrientes.
3. Ejecutar scripts de emplazamiento de cajas.
4. Ejecutar scripts correspondientes a cada tipo de entubado.
5. Generar alcances tridimensionales y bidimensionales en cuanto al entubado generado.

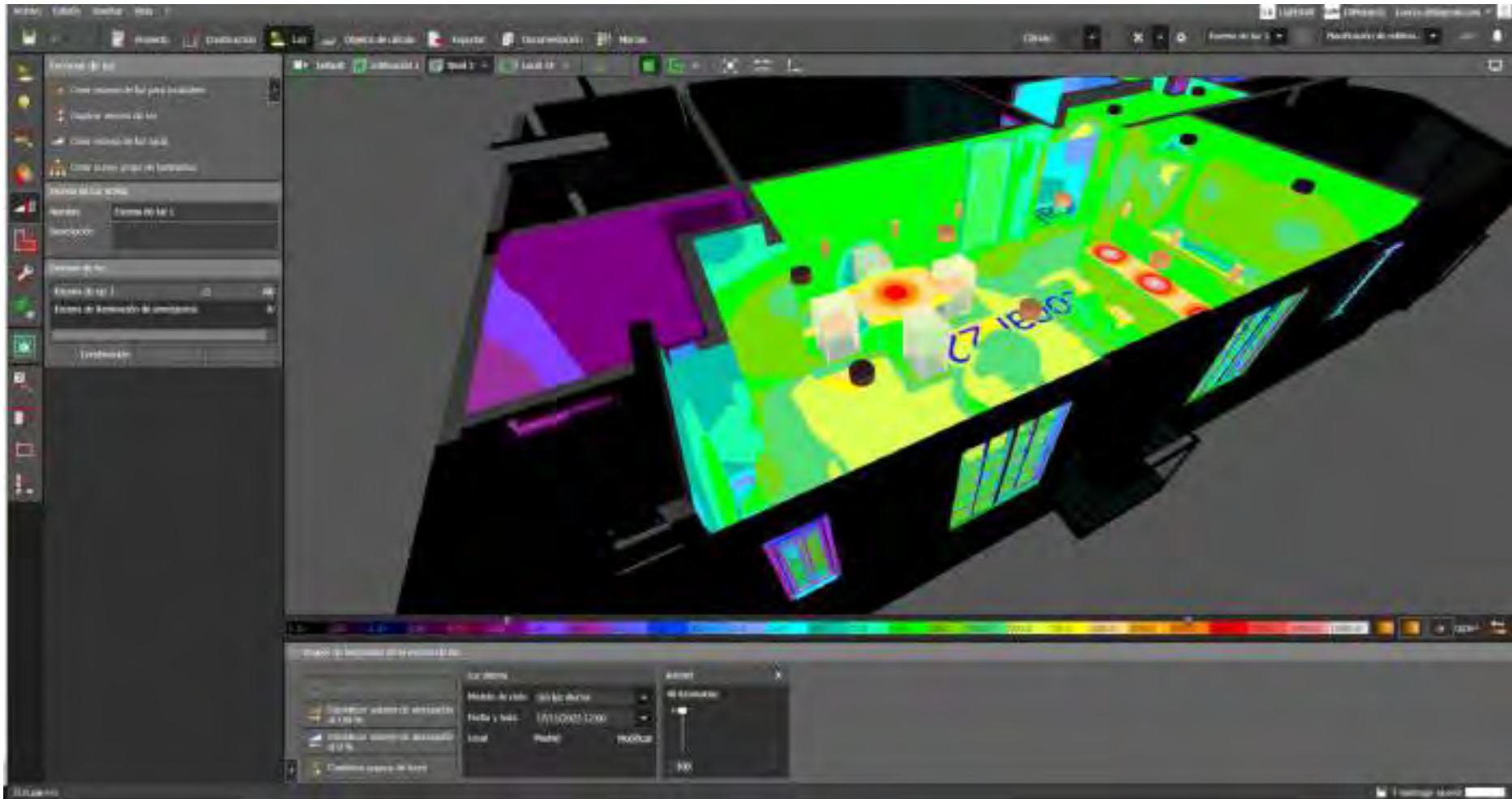
Para este nuevo caso se recurre el aspecto aplicativo a las plantas destinadas a uso de vivienda. Para lograr esto se debe entender que ya se posee un emplazamiento previo, tanto de luminarias, tomacorrientes u otras cargas que se requieran.

Por ejemplo, en la Figura 6.16 se muestra la interacción directa con Dialux EVO para la propuesta de diseños lumínicos y vistas que cumplan con los referenciales de la norma. Es importante considerar que en espacios privados destinados a viviendas, se deja espacio abierto a la subjetividad y a los gustos del propietario para el diseño lumínico. Esta interacción se da dentro del flujo de trabajo inicial con los scripts previamente desarrollados, los cuales ayudarán en el nuevo emplazamiento.

Del mismo modo, en el caso de los circuitos de tomacorrientes, se puede lograr una mayor rapidez en el trabajo al recurrir a las herramientas que el software BIM ofrece para crear rutas automáticas para los circuitos (ver Figuras 6.17 y 6.18).

**Figura 6.16**

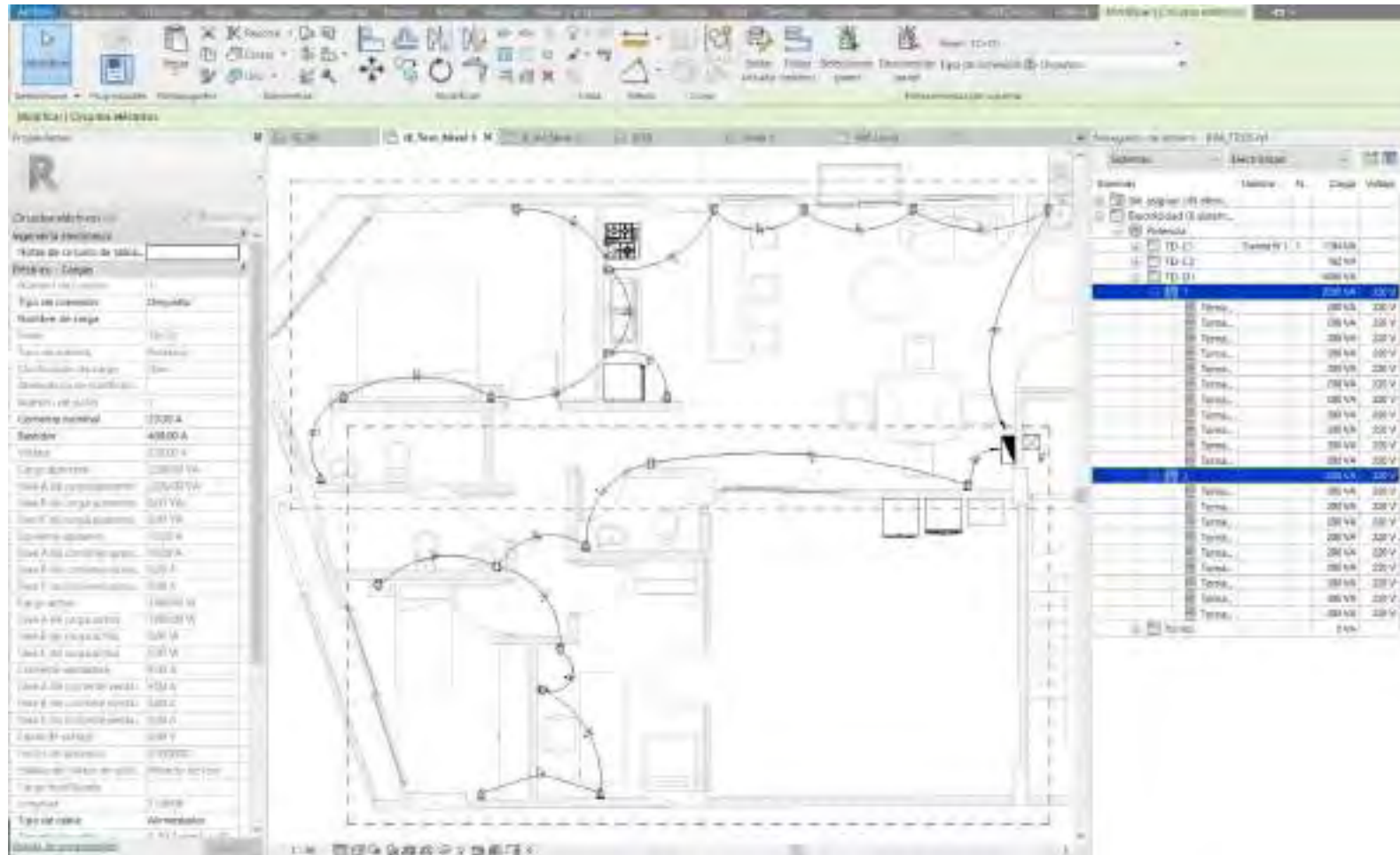
*Modelo de Dialux con diseño lumínico para ser reutilizado en Software BIM*



*Fuente:* Captura de software propia.

Figura 6.17

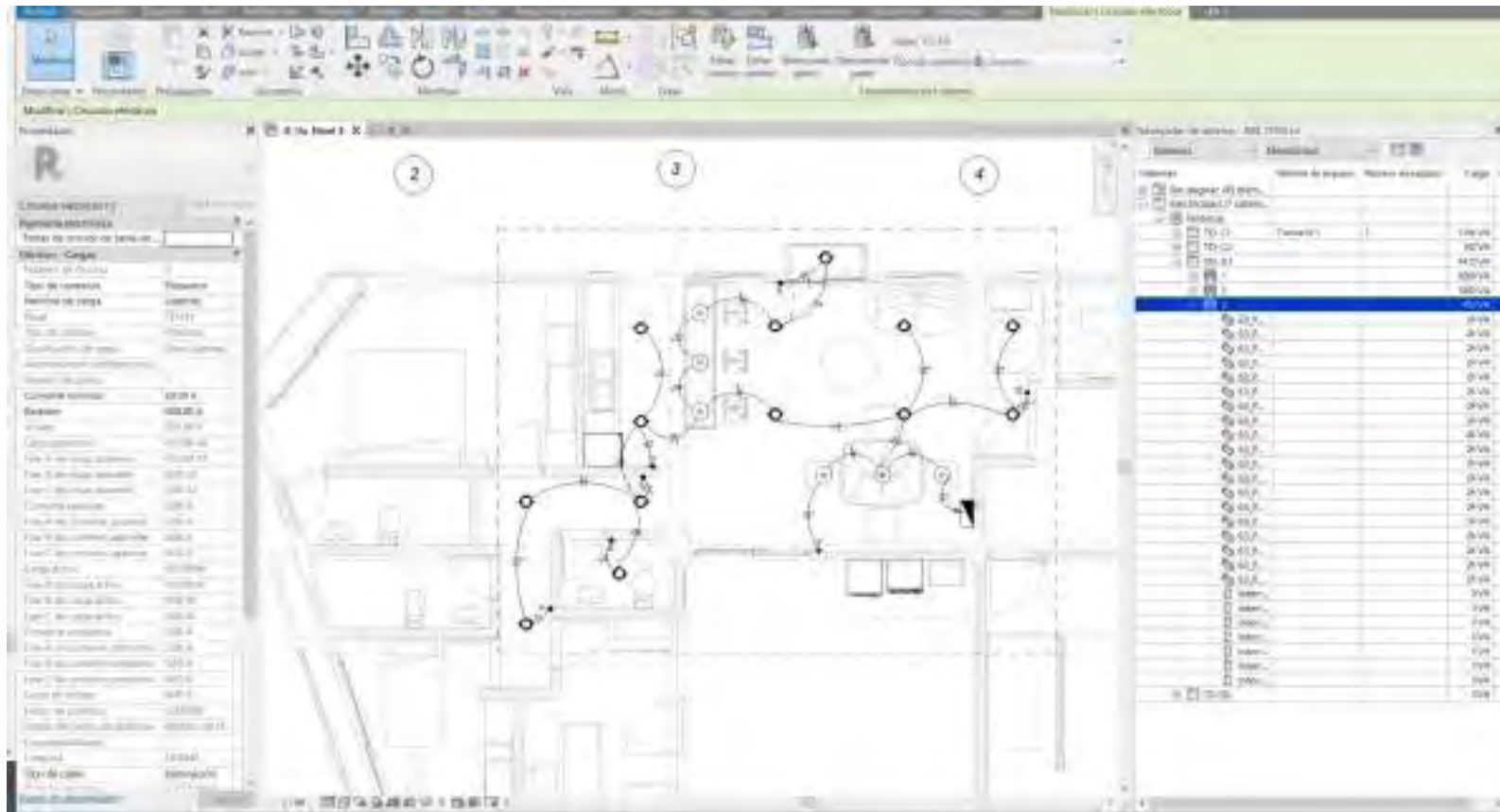
Planteado Automático de Circuitos de Tomacorrientes por medio de herramientas del entorno Electrical del Software BIM



Fuente: Captura de software propia.

Figura 6.18

Planteado Automático de Circuitos de Alumbrado por medio de herramientas del entorno Electrical del Software BIM



Fuente: Captura de software propia.



Con el escenario previo establecido, se puede proceder a la ejecución de scripts destinados al modelado de entubado, comenzando con una propuesta de montante eléctrica automatizada desde el medidor hasta un punto de tablero. En el primer proceso, como se muestra en la Figura 6.19, se exhibe una interfaz de selección sencilla pero poderosa en su funcionamiento, ya que permite generar subidas y bajadas de los conductos tanto en el mismo nivel como en diferentes niveles (Ver Figura 6.20).

**Figura 6.19**

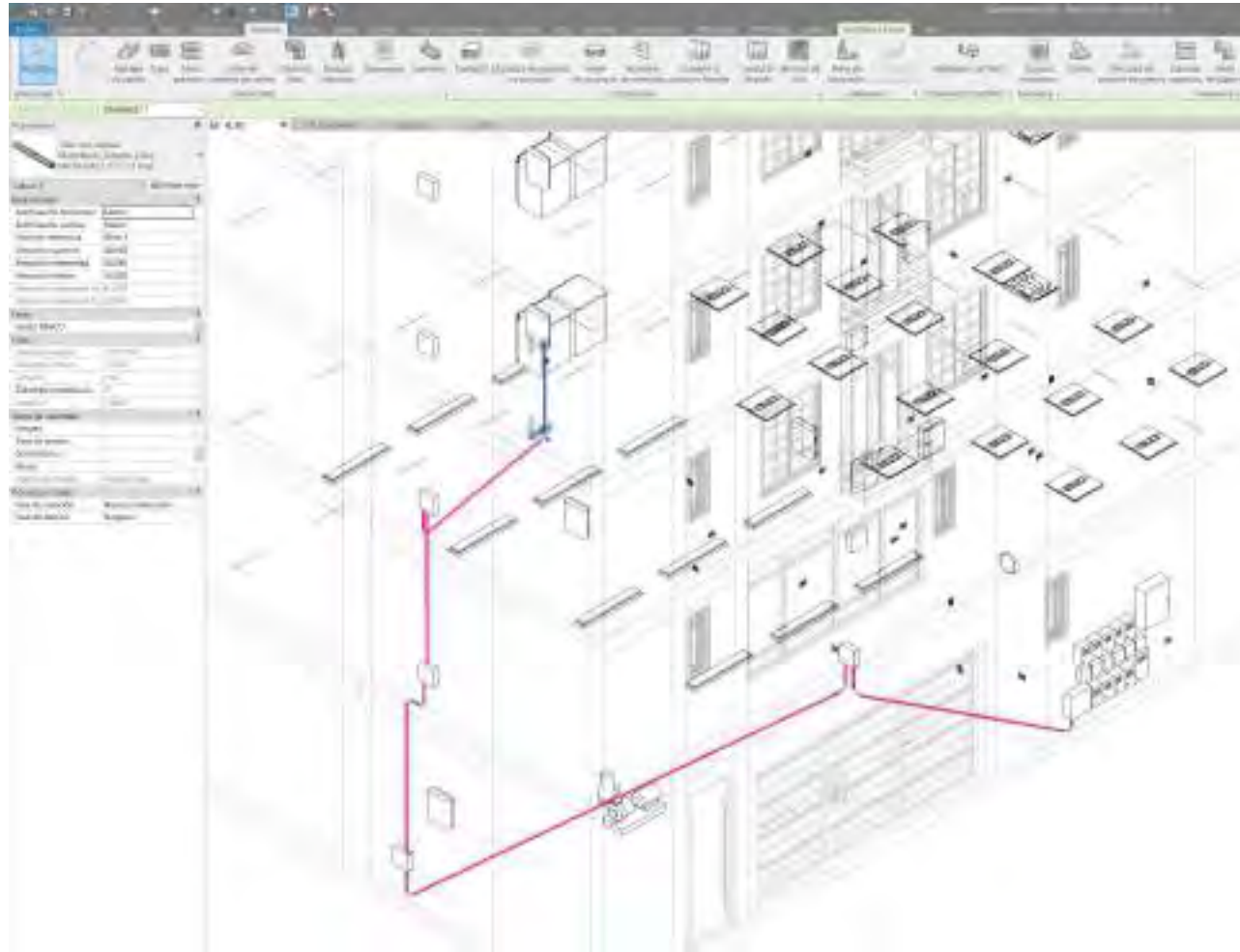
*Interfaz de ejecución de script para generación de conductos de alimentadores.*



*Fuente:* Captura de software propia.

**Figura 6.20**

*Generación automática de entubado y montante eléctrica*



*Fuente:* Captura de software propia.

En cuanto al tiempo de ejecución, considerando que el diseño propuesto en la aplicación ha recorrido cinco (5) cajas de paso, un (1) tablero de distribución y una montante que sube del primer al tercer nivel, se obtuvieron los siguientes resultados, como se muestra en las figuras 6.21 y 6.22.

Para el presente caso dentro de la interfaz de *Dynamo*, nuevamente se pueden conocer los tiempos de ejecución en donde se registraron los siguientes tiempos:

$$t_{ejecución} = 16\,787\text{ ms}$$

A su vez, tomando el criterio visto en la sección anterior de tomar en cuenta el registro de tiempo manual dentro de la operación se registra:

$$t_{Interfaz1} = 5632\text{ ms}$$

$$t_{Interfaz2} = 10\,442\text{ ms}$$

Resultando finalmente un tiempo de ejecución neto:

$$t_{Neto} = 713\text{ ms}$$

### Figura 6.21

*Tiempo de ejecución del script "Conductos-I"*



*Fuente:* Captura de software propia.

Figura 6.22

Tiempo de ejecución por nodo del script "Conductos-I"

ID	Nombre	Execution Time (ms)
76	if	0
77	Unit.FilterBySwathMask	0
16	Code Block	0
15	Code Block	1
54	Unit.ByStartFromCircleLength	1
17	GeometryType	1
23	Code Block	1
32	Code Block	1
42	Code Block	1
36	Point2	1
23	if	1
41	Code Block	1
39	Point2	1
56	Unit.Join	1
48	Point2	1
18	Code Block	1
57	Geometry.Translate	1
46	Curve.EndPoint	1
35	Unit.FilterBySwathMask	1
25	Code Block	2
45	Unit.FilterBySwathMask	2
34	Code Block	4
61	Unit.FilterBySwathMask	5
6	Feature Data	1
51	Python Script	11
63	Python Script	11
58	Python Script	11
78	Python Script	11
6	SelectMultipartToData	18
23	Clear List	18
29	Clear List	18
64	Python Script	20
43	Python Script	28
14	ElementCoordinates	38
66	Geometry	40
19	ConnectorLocation	48
11	ConnectorInCoordinate	69
22	ConnectorDirection	84
66	ConnectorByLines	102
63	FilterByMultipart	100
9	MultiplesFormas	1672
14	W_SelectElementosCycle	16442
Execution Time On Current Run		16757

Fuente: Captura de software propia.

Agregando funcionalidad, para los circuitos de alumbrado se preve una ejecución conjunta con lo propuesto en el script *Iluminación Óptima*, ya que a partir de este punto se pueden emplazar cajas de conexión para luego pasar a un proceso de entubado como se puede observar en la Figura 6.23.

### Figura 6.23

*Interfaz agregada para la inclusión de cajas de conexión en circuitos de alumbrado.*



*Fuente:* Captura de software propia.

Para este caso en particular, teniendo como base previa el análisis y estudio lumínico de las necesidades, que se pudo realizar según lo visto en la Figura 6.16, el proceso se simplifica enormemente. Ahora se puede seleccionar fácilmente el elemento BIM correspondiente a la caja de conexión a emplazar (Figura 6.24).

La ventaja de brindar una interfaz más amplia radica en que se pueden aprovechar las ubicaciones existentes para emplazar las cajas de conexión, lo que a largo plazo podría facilitar el control de cotas dentro de sistemas constructivos que lo requieran.

En cuanto al tiempo de ejecución, las figuras 6.26 y 6.27 muestran un detallado temporal que también tiene en cuenta la interacción manual del usuario al utilizar las interfaces para el emplazamiento de once (11) unidades de alumbrado, cada una con su respectiva caja de conexión o empalme.

**Figura 6.24**

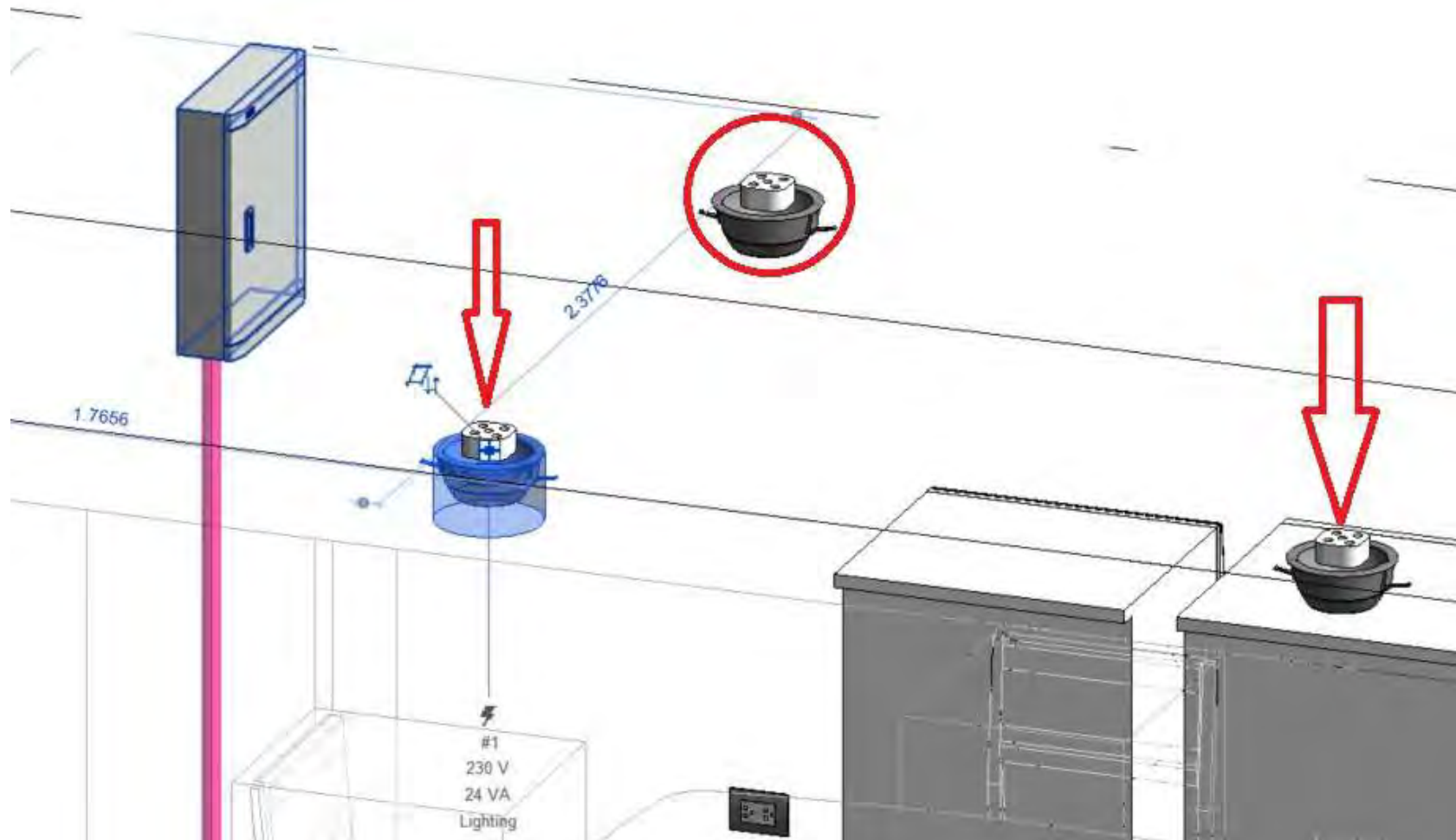
*Interfaz agregada para la selección de tipo de caja de conexión en circuitos de alumbrado.*



*Fuente:* Captura de software propia.

**Figura 6.25**

*Emplazamiento de cajas de conexión precisadas por coordenadas precisas de luminarias.*



*Fuente:* Captura de software propia.

El detalle de los tiempos de ejecución se exponen a continuación:

$$t_{ejecución} = 24\,449 \text{ ms}$$

Del anterior resultado se extrae a su vez el tiempo dedicado en interfaz :

$$t_{Interfaz1} = 5142 \text{ ms}$$

$$t_{Interfaz2} = 18\,799 \text{ ms}$$

Resultando finalmente un tiempo de ejecución neto:

$$t_{Neto} = 508 \text{ ms}$$

### Figura 6.26

*Tiempo de ejecución del script “Iluminación Óptima” más emplazamiento de cajas de conexión*



*Fuente:* Captura de software propia.



Figura 6.27

Tiempo de ejecución por nodo del script “Iluminación Óptima” más emplazamiento de cajas de conexión

ID	Node	Execution Time (ms)
46	Code Block	1
74	List GridItemIndices	1
42	List GridItemIndices	1
85	List GridItemIndices	1
50	Code Block	1
70	Code Block	1
77	—	1
43	Code Block	1
59	Code Block	1
60	List Color	1
47	List ColorMaterialIndex	1
25	Code Block	1
84	String ToString	1
07	String ToString	1
80	List Filter By Bool Mask	1
48	File From Path	1
81	String ToString	1
52	String Split	1
86	List Filter By Bool Mask	1
83	List Filter By Bool Mask	1
79	List GridItemIndices	1
78	Code Block	1
36	Level Get All Level	5
34	Select Faces Data	6
28	Textnote Data	6
22	Boolean Input	6
7	o] Family By Category	8
88	Point By Coordinates	8
37	DropDown Data	8
58	Select Made Elements Data	8
5	o] Family By Category	8
7	DropDown Data	9
34	File Path Data	15
62	DropDown Data	17
42	DropDown Data	23
73	Level Dynamic	54
17	ID Based Current Path	55
31	All Family Instances By Face McPoint	68
41	All Family Types Of Category	85
78	MultipleInputForm ++	5142
86	MultipleInputForm ++	18799
Execution Time On Current Run:		24449

Fuente: Captura de software propia.

Continuando con el proceso BIM descrito anteriormente, se procede a culminar con el emplazado de cajas de conexión, en este caso, cajas de tipo rectangulares. La operación del script es similar a los casos anteriores en cuanto a su interfaz (ver Figura 6.28), y los resultados pueden considerarse mejores debido a que, en el proceso de programación, se estableció una selección múltiple de anfitriones donde las cajas de conexión se puedan ubicar.

**Figura 6.28**

*Interfaz para el emplazamiento de cajas rectangulares.*



*Fuente:* Captura de software propia.

Es esta misma interfaz la cual nos permite un manejo masivo de elementos. Para el caso aplicativo actual se considera un registro de tiempo de ejecución para el emplazado de cajas de un circuito de tomacorrientes de once (11) unidades emplazadas según propuesta arquitectónica. Tales registros son:

$$t_{ejecución} = 29\,744 \text{ ms}$$

Del anterior resultado se extrae a su vez el tiempo dedicado en interfaz :

$$t_{Interfaz1} = 29\,536 \text{ ms}$$

Resultando finalmente un tiempo de ejecución neto:

$$t_{Neto} = 208 \text{ ms}$$

**Figura 6.29**

*Tiempo de ejecución del script “Tomacorrientes” más emplazamiento de cajas de conexión*



*Fuente:* Captura de software propia.

**Figura 6.30**

*Tiempo de ejecución por nodo del script “Tomacorrientes” más emplazamiento de cajas de conexión*

Force Re-execute		Total Graph Execution Time: 29744ms
#	Name	Duration (ms)
<b>Executing</b>		
	Categories	0
	Categories	0
<b>Executed On Current Run</b>		
8	TimDimInstancesByCategoryPart	34
7	Code Block	0
6	UnHidden	0
5	Code Block	1
4	GeometryControlTo	6
3	Element.GetLocation	0
2	Code Block	0
1	Code Block	0
0	MultiplanarSection	0

*Fuente:* Captura de software propia.

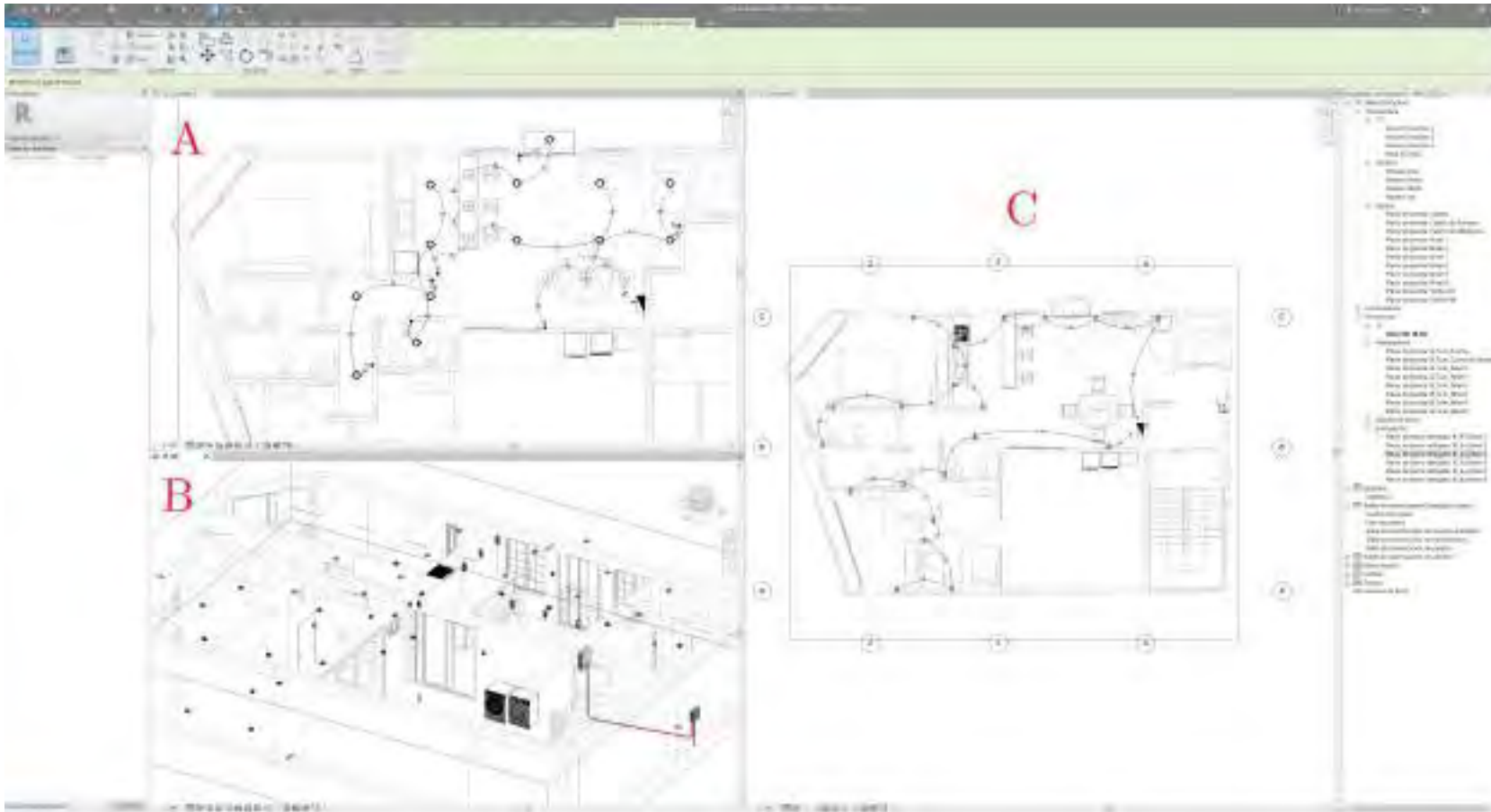
Los registros antes planteados pueden ser verificados en las figuras 6.29 y 6.30.

Finalmente, en el ejecutado del entubado es necesario recurrir a los scripts correspondientes a esta tarea, que dada la situación aplicativa de la presente tesis requiere de repetidas ejecuciones para completar las características totales por circuito eléctrico (ya sea de iluminación o alumbrado). En este sentido, las ventajas BIM y su software dedicado resaltan sus funcionalidades al tener la capacidad de manipulación de distintas vistas de un mismo proyecto con diferentes características, como se puede apreciar en la Figura 6.31.

El flujo de trabajo puede desprender un detalle de ser ejecutado bajo esta prerrogativa de tener múltiples vistas, ya que, en referencia a la Figura 6.31, se logran aprovechar vistas de circuitos planteados con anterioridad (A y C) para agilizar una rápida ejecución a través de la vista tridimensional (3D).

**Figura 6.31**

*Vistas múltiples de un mismo proyecto como apoyo a la ejecución de scripts dentro del software BIM.*

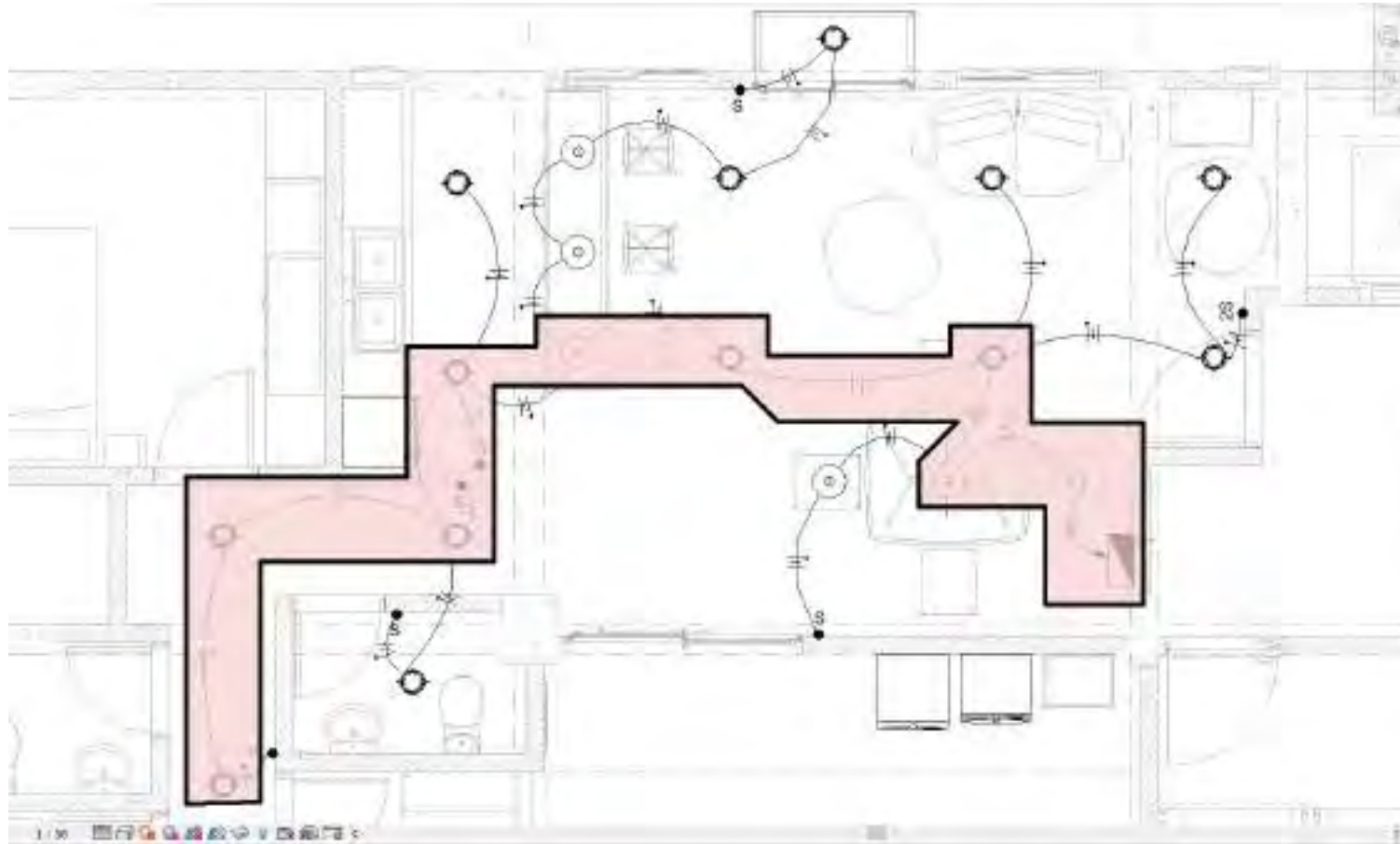


*Fuente:* Captura de software propia.

De este modo, se procede a entubar los elementos que sean necesarios en base al camino planteado por los circuitos. Para ello, es importante determinar una red “Troncal” (Figura 6.32a y Figura 6.32b) por la cual establecer los primeros entubados y luego continuar con los ramales correspondientes.

**Figura 6.32**

*Propuestas de zona troncal de circuitos de iluminación y tomacorrientes*

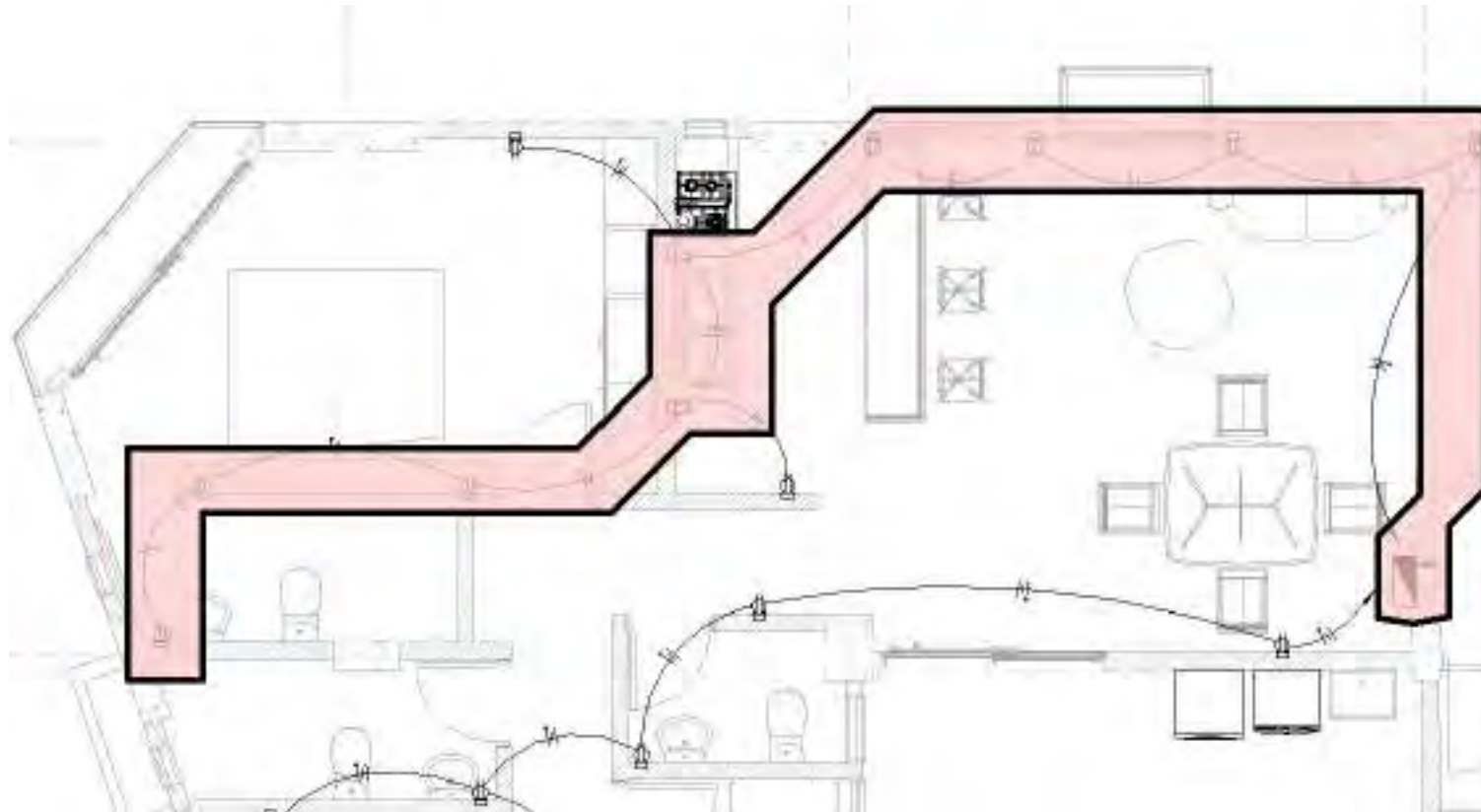


**(a)** *Circuito de Iluminación.*

*Fuente:* Captura de software propia.

**Figura 6.32**

*Propuestas de zona troncal de circuitos de iluminación y tomacorrientes (Continuación)*



*(b) Circuito de Tomacorrientes.*

*Fuente:* Captura de software propia.

En tanto a las interfaces correspondientes vistas en las figuras 6.33a y 6.33b, se presta similitud entre los dirigidos a alumbrado y tomacorrientes ya que comúnmente se dan por techo y suelo respectivamente.

### Figura 6.33

*Interfaz de ejecución del script para entubados por techo y suelo*



(a) *Interfaz de ejecución para entubado de circuitos por techo.*



(b) *Interfaz de ejecución para entubado de circuitos por suelo.*

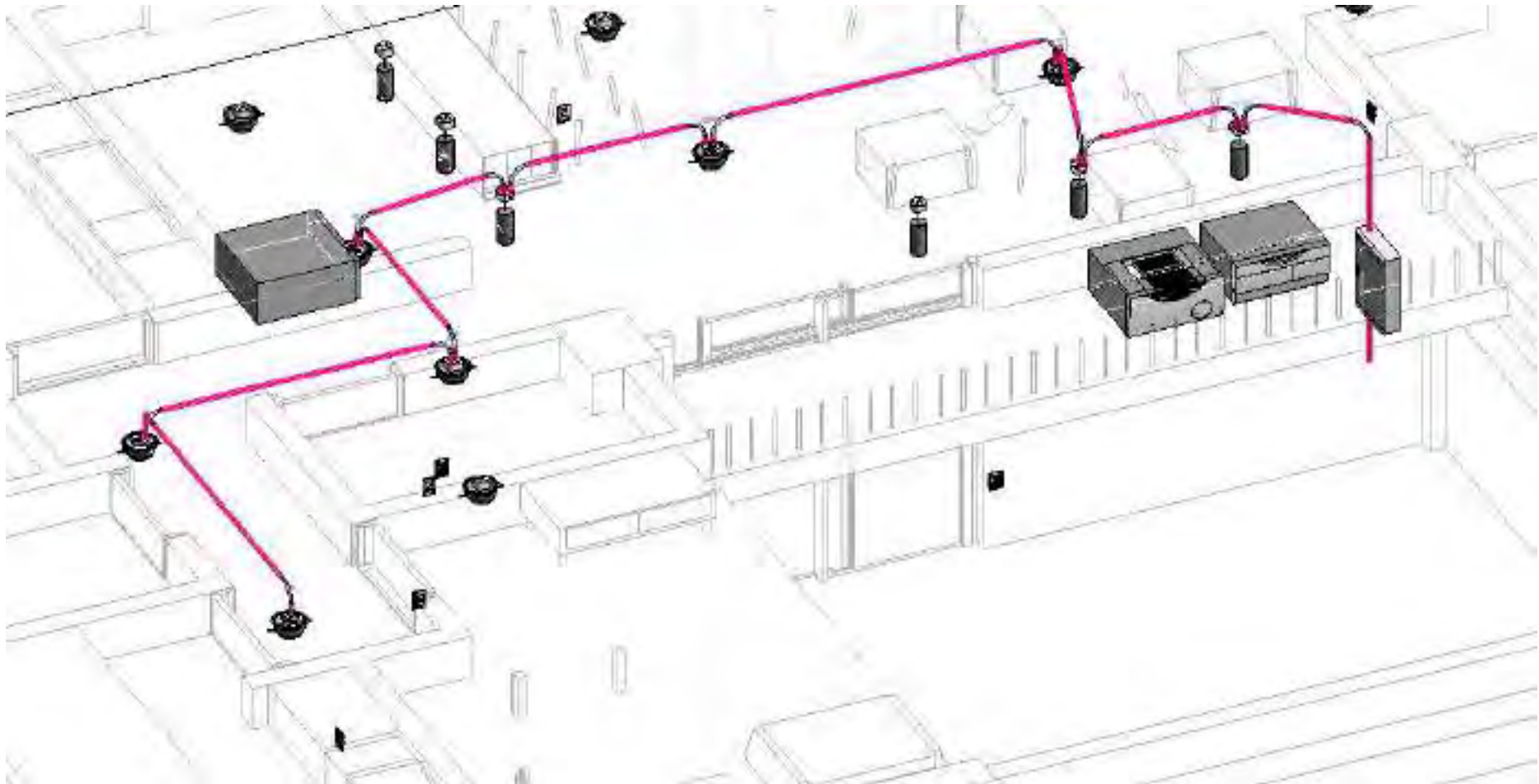
*Fuente:* Captura de software propia.

Tomando en cuenta la Figura 6.32a, se puede observar el resultado obtenido en la Figura 6.34, donde se muestra la generación automática del conducto respectivo para el circuito de alumbrado. Del mismo modo, para el circuito de tomacorrientes, basado en la Figura 6.32b, se puede apreciar la generación automática del conducto correspondiente en la Figura 6.35.



**Figura 6.34**

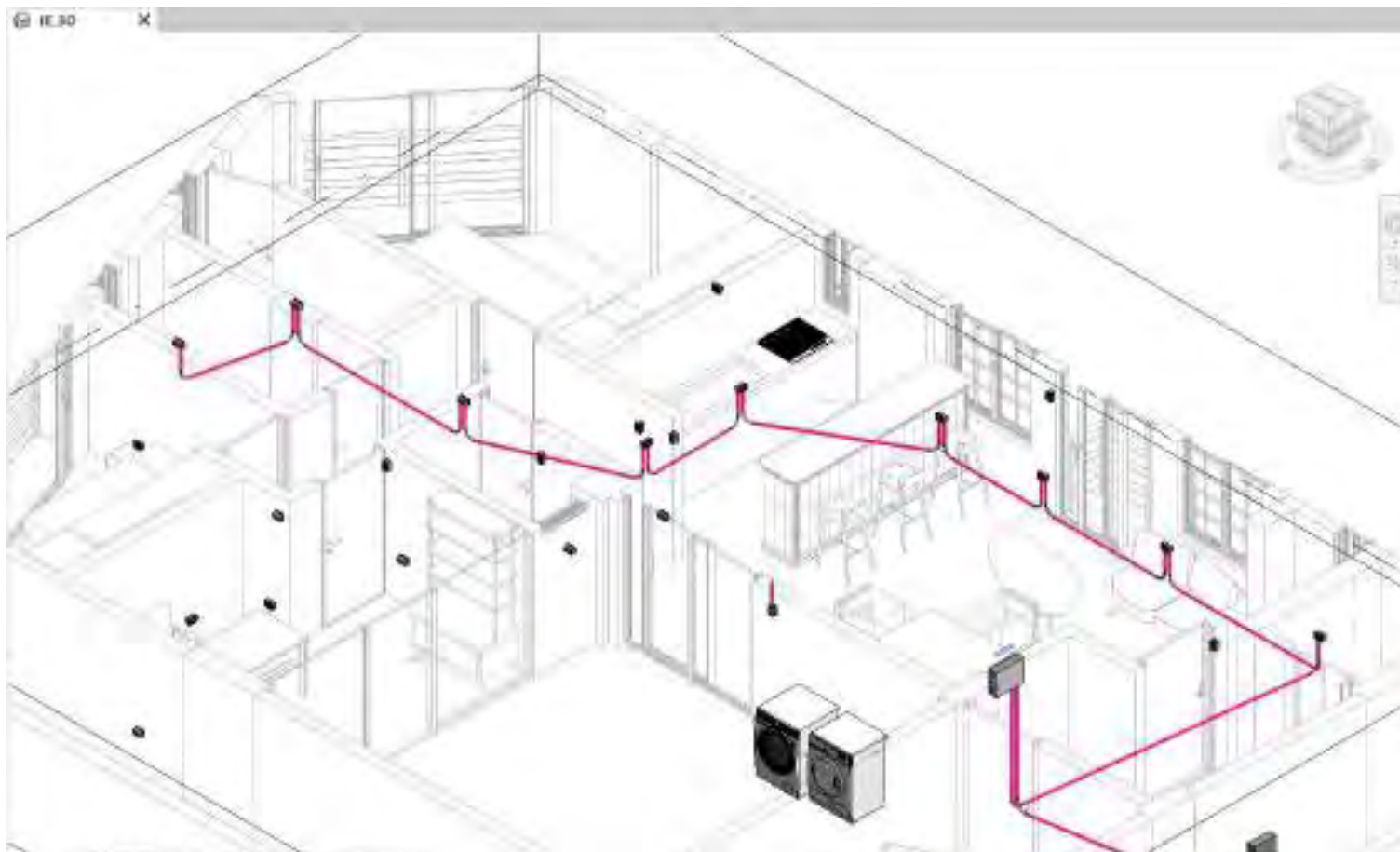
*Vista del entubado generado por el script de entubado por techo.*



*Fuente:* Captura de software propia.

**Figura 6.35**

*Vista del entubado generado por el script de entubado por suelo.*



*Fuente:* Captura de software propia.

El tiempo de ejecución para estas tareas se detalla a continuación:

#### 6.4.4.0.1 Entubado por Techo

$$t_{ejecución} = 29\,273 \text{ ms}$$

$$t_{Interfaz1} = 28\,418 \text{ ms}$$

$$t_{Neto} = 855 \text{ ms}$$

El tiempo neto de ejecución obtenido anteriormente (855 ms) se puede verificar en las figuras 6.36a y 6.37b, donde se toma en cuenta que se ha emplazado exactamente 11,538 m de conducto de tubería, incluyendo veintiún (21) curvas o codos correspondientes para su embonado.

La longitud de los conductos puede ser extraída directamente de la interfaz de *Revit* para el caso de las curvas; sin embargo, para los conductos se puede recurrir a una automatización por *Dynamo* para obtener la longitud, como se muestra en la Figura 6.38a.

#### 6.4.4.0.2 Entubado por Suelo

$$t_{ejecución} = 35\,154 \text{ ms}$$

$$t_{Interfaz1} = 32\,963 \text{ ms}$$

$$t_{Neto} = 2191 \text{ ms}$$

Los resultados obtenidos pueden ser verificados en las figuras 6.36a y 6.37a.

Se toma en consideración que para la ejecución de esta tarea se ha emplazado exactamente 23,046 m de conducto de tubería, incluyendo diecinueve (19) curvas o codos correspondientes para su embonado.

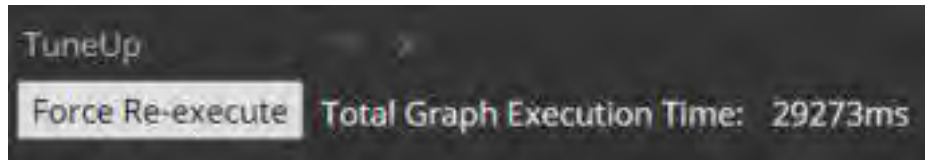
Al igual que en el caso anterior, se puede recurrir directamente a la interfaz de *Revit* para las curvas; mientras que para los conductos nuevamente se plantea el uso de *Dynamo* para obtener la longitud, como se muestra en la Figura 6.38b.

A partir de este proceso de aplicación, se puede extender el funcionamiento de los scripts desarrollados a los demás componentes de la instalación eléctrica. A lo largo de esta sección, se ha

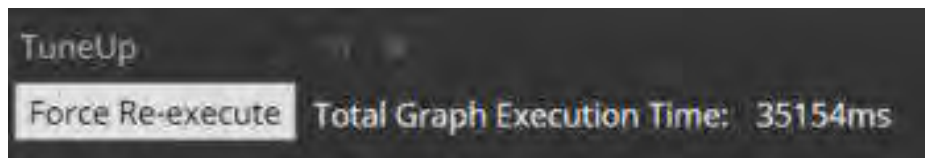
puesto a prueba el funcionamiento de los scripts con diferentes objetivos de modelado, enfrentando situaciones donde se requería un planteamiento más complejo y otras en las que las dificultades disminuían en la ejecución.

**Figura 6.36**

*Tiempo de ejecución por nodo de scripts de conductos por techo y suelo.*



(a) *Conductos por techo para un circuito troncal de iluminación.*



(b) *Conductos por suelo para un circuito troncal de tomacorrientes.*

*Fuente:* Captura de software propia.

Figura 6.37

Tiempo de ejecución por nodo de scripts de conductos por techo y suelo.

TuneUp		
Force Re-execute Total Graph Execution Time: 29273ms		
#	Name	Execution Time (ms)
62	Element Type	0
34	Code Block	1
41	Geometry Translate	1
52	Curve EndPoint	1
23	Code Block	1
25	Code Block	1
24	Code Block	1
3	Categories	1
27	Code Block	1
46	Point.2	1
50	Code Block	1
48	Point.1	1
47	Point.1	1
34	Point.2	1
42	List Filter by Boolean Mask	1
33	List Filter by Boolean Mask	1
57	List Create	1
56	List Create	1
45	Code Block	1
44	Code Block	1
66	List Flatten	1
60	Code Block	1
64	Code Block	1
40	#	1
35	Code Block	1
41	Code Block	1
16	DropDown Data	2
4	All Elements of Category	5
8	All Elements of Category	5
14	Select Model Elements Data	5
6	DropDown Data	10
18	Select Model Elements Data (ordered)	10
17	TextBox Data	10
7	Categories	14
51	List Filter by Boolean Mask	15
49	Python Script	39
36	Connector Direction	42
32	Connector as Connected	38
20	Connector Location	85
43	Element Level	37
38	Conduit by User	180
65	Elbow by MEP Curves	296
33	Multiple Input Form 1.1	29475
Execution Time On Current Run		29273

(a) Conductos por techo para un circuito troncal de iluminación

TuneUp		
Force Re-execute Total Graph Execution Time: 35154ms		
#	Name	Execution Time (ms)
8	Categories	0
18	List Join	1
60	List Create	1
5	Code Block	1
28	Code Block	1
30	Code Block	1
45	Code Block	1
50	Curve Start Point	1
47	Curve End Point	1
20	Code Block	1
40	Point.1	1
41	Point.2	1
53	Point.2	1
35	Code Block	1
40	List Filter by Boolean Mask	1
30	#	1
28	Code Block	1
48	List Flatten	1
51	List Flatten	1
50	Code Block	2
36	Vector.2	2
58	Geometry Translate	2
20	Element Type	3
37	Code Block	3
27	Code Block	4
32	List Filter by Boolean Mask	6
17	Select Model Elements Data	11
9	Categories	15
16	TextBox Data	17
44	Python Script	31
30	MFC Connections Get Connectors	39
4	Code Block	36
31	Connector as Connected	38
3	All Elements of Category	52
17	Select Model Elements Data (ordered)	51
35	Connector Direction	62
34	Connector Location	66
25	Element Level	90
62	Conduit by User	180
65	Elbow by MEP Curves	257
6	DropDown Data	1150
15	Multiple Input Form 1.1	32154
Execution Time On Current Run		35154

(b) Conductos por suelo para un circuito troncal de tomacorrientes.

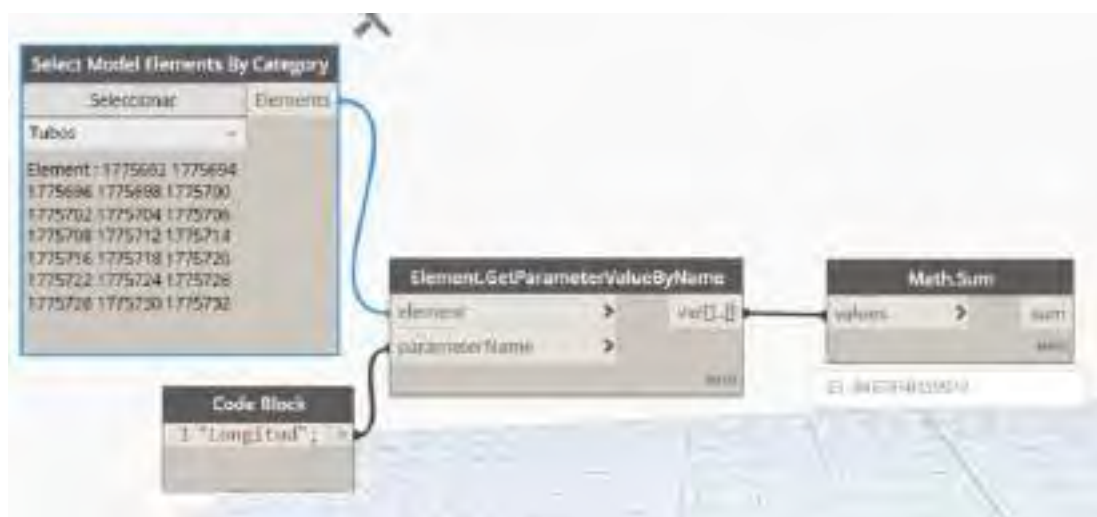
Fuente: Captura de software propia.

Figura 6.38

Aplicación de Dynamo para la toma de longitudes de conductos generados.



(a) Circuitos por techo.



(b) Circuitos por suelo.

Fuente: Implementación propia.

### 6.4.5 Ejecución de Cálculos

En esta etapa de la aplicación de la propuesta, se configura la parte final del proceso de diseño y cálculo de instalaciones eléctricas interiores. Con los detalles completos del modelo BIM que servirá como base para los cálculos, es posible proceder a ejecutar los programas de la propuesta y automatizar los cálculos necesarios para documentar adecuadamente el proyecto de instalaciones eléctricas.

#### 6.4.5.1 Planteamiento de Cuadro de Cargas

Utilizando las herramientas BIM incorporadas en el software donde se realizó el modelado previo, es viable obtener datos precisos de área que facilitan la pronta ejecución de los scripts programados (ver Figura 6.39).

**Figura 6.39**

*Extracción de datos de áreas por medio de herramientas BIM.*



*Fuente:* Captura de software propia.

En el contexto de la presente aplicación, se establece que el script llevará a cabo los cálculos de carga básica para los dos tipos de edificaciones principales presentes en el proyecto: Espacios Comerciales y Espacios de Vivienda.

La ejecución del script se inicia mediante la interfaz de entrada de datos (véase Figuras 6.40 y 6.41), donde se selecciona en primer lugar el tipo específico de edificación para el cual se realizará el cálculo de cargas.

**Figura 6.40**

*Interfaz de generación de cuadro de cargas basados en tipos de edificación de la Sección 050 del CNE-U*



*Fuente:* Captura de software propia.

**Figura 6.41**

*Interfaz de generación de cuadro de cargas basados en tipos de edificación de la Tabla N° 14 del CNE-U*



*Fuente:* Captura de software propia.



La propuesta desarrollada ha sido diseñada para ser altamente programable, lo que brinda flexibilidad en los cálculos realizados y en la consideración de normativas específicas. Siguiendo los lineamientos del CNE-U, se solicitan uno o dos datos de área, dependiendo del tipo de edificación, y se ofrece la opción de incorporar cargas adicionales que puedan ser relevantes para el cálculo.

En esta línea, la interfaz presentada en la Figura 6.42 permite una selección rápida de elementos dentro del software BIM, según su clasificación o uso conforme al CNE-U. Esta funcionalidad facilita la inclusión precisa de las cargas adicionales en el cálculo.

**Figura 6.42**

*Interfaz para la toma de datos de área y posibles cargas adicionales*



*Fuente:* Captura de software propia.

En consecuencia, se propone la generación de cuadros de carga tanto para los espacios comerciales como para los domésticos. A partir de este punto, se abre la posibilidad de evaluar la versatilidad del sistema, permitiendo la inclusión opcional de cargas adicionales en casos donde la normativa establezca tratamientos especiales.

**6.4.5.1.1 Local Comercial** A continuación, se presentan los tiempos de ejecución obtenidos para un solo local comercial, considerando únicamente la carga básica de acuerdo con las especificaciones del CNE-U:

$$t_{ejecución} = 15\,682 \text{ ms}$$

$$t_{Interfaz1} = 551 \text{ ms}$$

$$t_{Interfaz2} = 3384 \text{ ms}$$

$$t_{Interfaz3} = 4421 \text{ ms}$$

$$t_{Interfaz4} = 4715 \text{ ms}$$

$$t_{Neto} = 2611 \text{ ms}$$

Los tiempos mencionados anteriormente corresponden a la generación del cuadro dentro de la misma interfaz de Revit. Esto permite que el cuadro de cargas pueda ser manipulado directamente en el entorno del software BIM, eliminando la necesidad de recurrir a software de terceros para importar tablas o cuadros.

**Figura 6.43**

*Cuadro de Cargas generado para un local comercial*

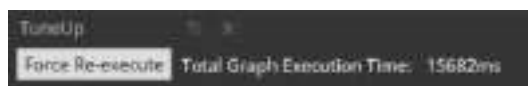
	A	B	C	D	E	F	G
Nombre de la carga	Punto	Carga	Descripción	Frecuencia	PTI	Módulo de carga	
1		CARGA BARCA	20 BARCOS CUBIERTOS	12K.25			12K.25
2			TOTAL	12K.25			12K.25
3		CARGA AEROSPA					
4				0			0
5				0			0
6		APL. DE FACTORES					
7		ACORREDA		12K.25	11		12K.25
8		ADJUSTADOR		12K.25	11		12K.25
9			TOTAL	12K.25			12K.25

*Fuente:* Captura de software propia.

A su vez, a continuación se observan los datos obtenidos desde la interfaz de *Dynamo*.

**Figura 6.44**

*Tiempo de ejecución del script "Cuadro de Cargas"*



*Fuente:* Captura de software propia.

Figura 6.45

Tiempo de ejecución por nodo del script “Cuadro de Cargas”

#	Name	Execution Time (ms)
159	if	19
225	Code Block	19
200	Code Block	19
218	Return	20
258	Code Block	20
212	ScheduleView/ScheduleFields	20
169	Code Block	20
192	Code Block	20
248	Code Block	21
221	Code Block	22
236	Code Block	22
256	Code Block	22
215	App / dem.	22
228	Code Block	23
42	Code Block	23
240	Code Block	24
125	Get json	25
196	Code Block	25
210	Code Block	25
195	Input Field Data	26
243	Python Script	27
68	TreeView Input	27
145	Python Script	28
234	Python Script	28
183	Python Script	28
189	Python Script	28
173	Python Script	28
102	Python Script	30
264	Python Script	31
64	Select.ByCategoryAndLevel	31
39	Select.ByCategoryAndLevel	31
262	Python Script	31
118	Python Script	31
186	Python Script	31
271	Python Script	31
57	o) Element by category	38
275	...	40
118	MultipleInputDialog ++	551
271	key Schedule	879
91	MultipleInputDialog ++	1384
85	MultipleInputDialog ++	4421
105	MultipleInputDialog ++	8715
	Execution Time On Current Run	15682

Fuente: Captura de software propia.



Figura 6.48

Tiempo de ejecución por nodo del script “Cuadro de Cargas”

TuneUp  
Force Re-execute Total Graph Execution Time: 28832ms

#	Name	Execution Time (ms)
	Code Block	21
	Code Block	22
	Code Block	22
	Code Block	22
	Code Block	23
	Code Block	23
	Code Block	23
	Code Block	23
	Code Block	24
	Code Block	24
	Code Block	24
	Code Block	24
	Code Block	24
	Code Block	25
	Code Block	26
	List Create	26
	Code Block	26
	Code Block	26
	ScheduleView.CreateSchedule	27
	TextBox Data	27
	Code Block	28
	List Flatten	28
	Python Script	29
	Code Block	30
	Code Block	30
	Clear List	31
	Python Script	32
	Python Script	32
	Python Script	32
	Python Script	33
	Select.ByCategoryAndLevel	33
	Python Script	33
	Clear List	34
	Python Script	35
	Select.ByCategoryAndLevel	35
	Clear List	36
	==	45
	o  Element.ByCategory	45
	MultipleInputForm ++	100
	MultipleInputForm ++	3660
	MultipleInputForm ++	9904
	MultipleInputForm ++	12299
	Execution Time On Previous Run	27903

Fuente: Captura de software propia.

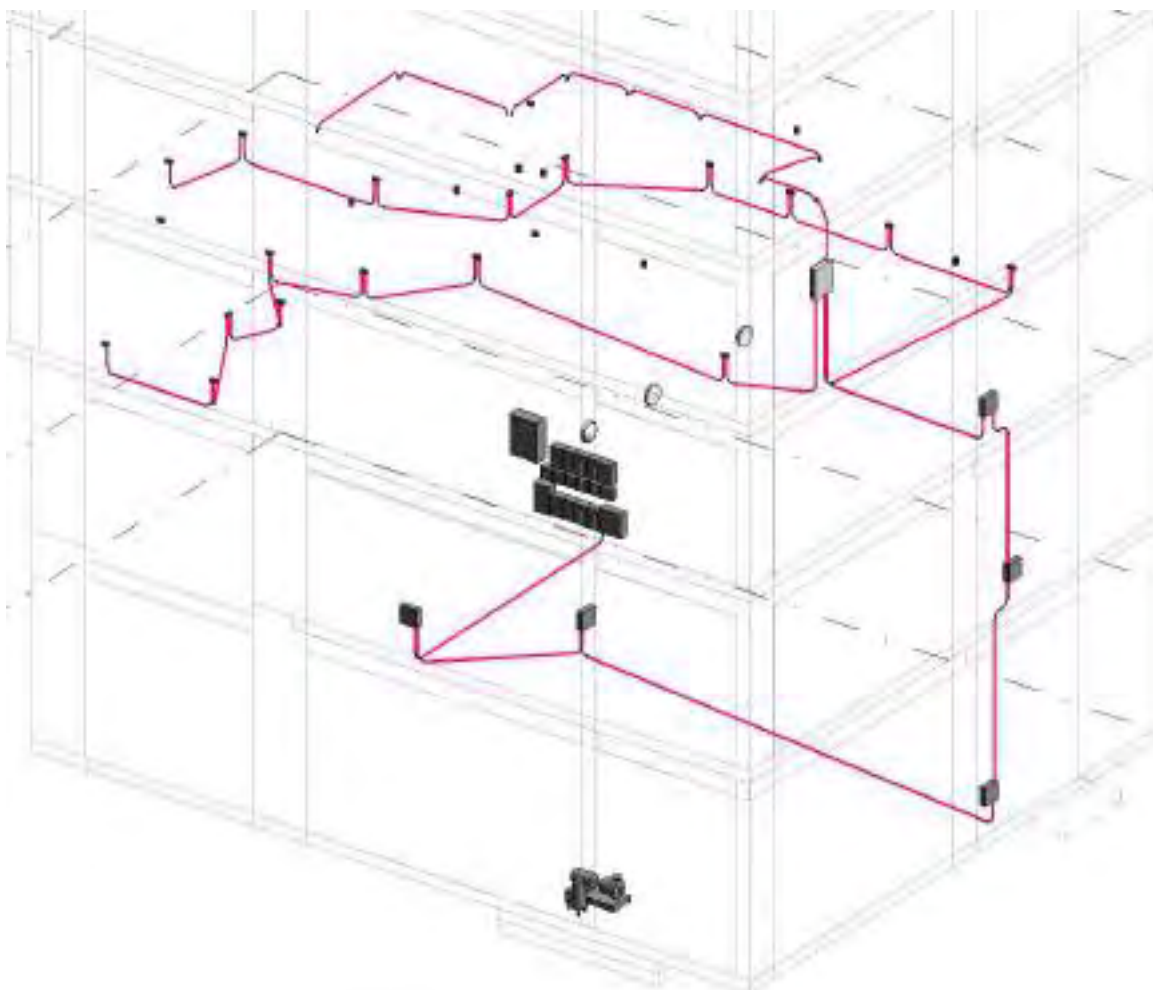
### 6.4.5.2 Cálculo de Caídas de Tensión

Para la ejecución del script dedicado al cálculo de caídas de tensión se debe tener en cuenta que, como punto inicial, es preferible manejar las vistas tridimensionales del modelo BIM en el que se está trabajando.

Para el particular caso, inicialmente el entubado troncal se puede manejar bajo otros colores, esto con el objeto de poder interpretar de mejor manera las caídas de tensión que se obtenga.

#### Figura 6.49

*Modelo BIM del entubado para un tablero de distribución previo a la ejecución del script de cálculo de caídas de tensión*



*Fuente:* Captura de software propia.

Además, es importante considerar el aspecto mencionado anteriormente con respecto al entorno “Electrical”, que se refiere al cálculo automático de los calibres de los conductores. En la Figura 6.50, se puede observar el cálculo realizado por Revit para un circuito, mientras que más adelante se presenta un parámetro adicional creado por el usuario que permite su manipulación. En

los próximos puntos se detallarán las diferencias en los resultados entre estos dos enfoques, lo cual puede ser muy relevante para el funcionamiento general de la instalación eléctrica.

**Figura 6.50**

*Diferencia de parámetros del calibre de conductor*

Circuitos eléctricos (1)	
Ingeniería eléctrica	
Notas de circuito de tabla de plan...	
Eléctrico - Cargas	
Número de circuito	A-1
Tipo de conexión	Disyuntor
Nombre de carga	TD-D1
Ícono	IFE-A-TD-D1
Tipo de sistema	Potencia
Clasificación de carga	Dim Lighting
Abreviatura de clasificación de cá...	
Número de punto	1
Corriente nominal	2000 A
Bestidor	400.00 A
Voltaje	220.00 V
Carga aparente	44121.2 VA
Fase A de carga aparente	14707.1 VA
Fase B de carga aparente	0.00 VA
Fase C de carga aparente	0.00 VA
Corriente aparente	200.0 A
Fase A de corriente aparente	200.0 A
Fase B de corriente aparente	0.00 A
Fase C de corriente aparente	0.00 A
Carga activa	40230.0 W
Fase A de carga activa	40230.0 W
Fase B de carga activa	0.00 W
Fase C de carga activa	0.00 W
Corriente reactiva	18.42 A
Fase A de corriente reactiva	18.42 A
Fase B de corriente reactiva	0.00 A
Fase C de corriente reactiva	0.00 A
Caída de voltaje	0.00 V
Factor de potencia	0.918506
Espacio del factor de potencia	Retardo de fase
Carga equilibrada	
Longitud	15.8024
Tipo de cable	Alimentador
Tamaño de cable	1-#2.5 mm <sup>2</sup> , 1-#2.5 mm <sup>2</sup> , 1-#12
Núm. de ramas	1
Núm. de contactos regulables	1
Núm. de contactos regulador	1
Núm. de conductores de tierra	1
Número de interruptor	1
Tamaño de cable Corregido	1-#6 mm <sup>2</sup> , 1-#6 mm <sup>2</sup>
Lighting corrector	433.00 VA
De a Colectado	4000.00 VA

*Fuente:* Captura de software propia.

Con este contexto establecido, la interfaz que marca el inicio de la tarea de cálculo de caídas de tensión es la que se presenta en la Figura 6.51.

A partir de este punto, la obtención de los resultados es completamente automatizada. Se debe tener en claro que se obtendrán dos resultados bien diferenciados. En primer lugar, se generará el cuadro de caídas de tensión como se muestra en la Figura 6.52. En segundo lugar, se generará

**Figura 6.51**

*Interfaz para el cálculo de caídas de tensión*



*Fuente:* Captura de software propia.

la vista tridimensional con una representación de colores que indica los niveles de caídas de tensión, como se aprecia en la Figura 6.53. Estos resultados proporcionan una interpretación visual de las caídas de tensión en la instalación eléctrica.

**Figura 6.52**

*Cuadro de Caídas de Tensión generado por script*

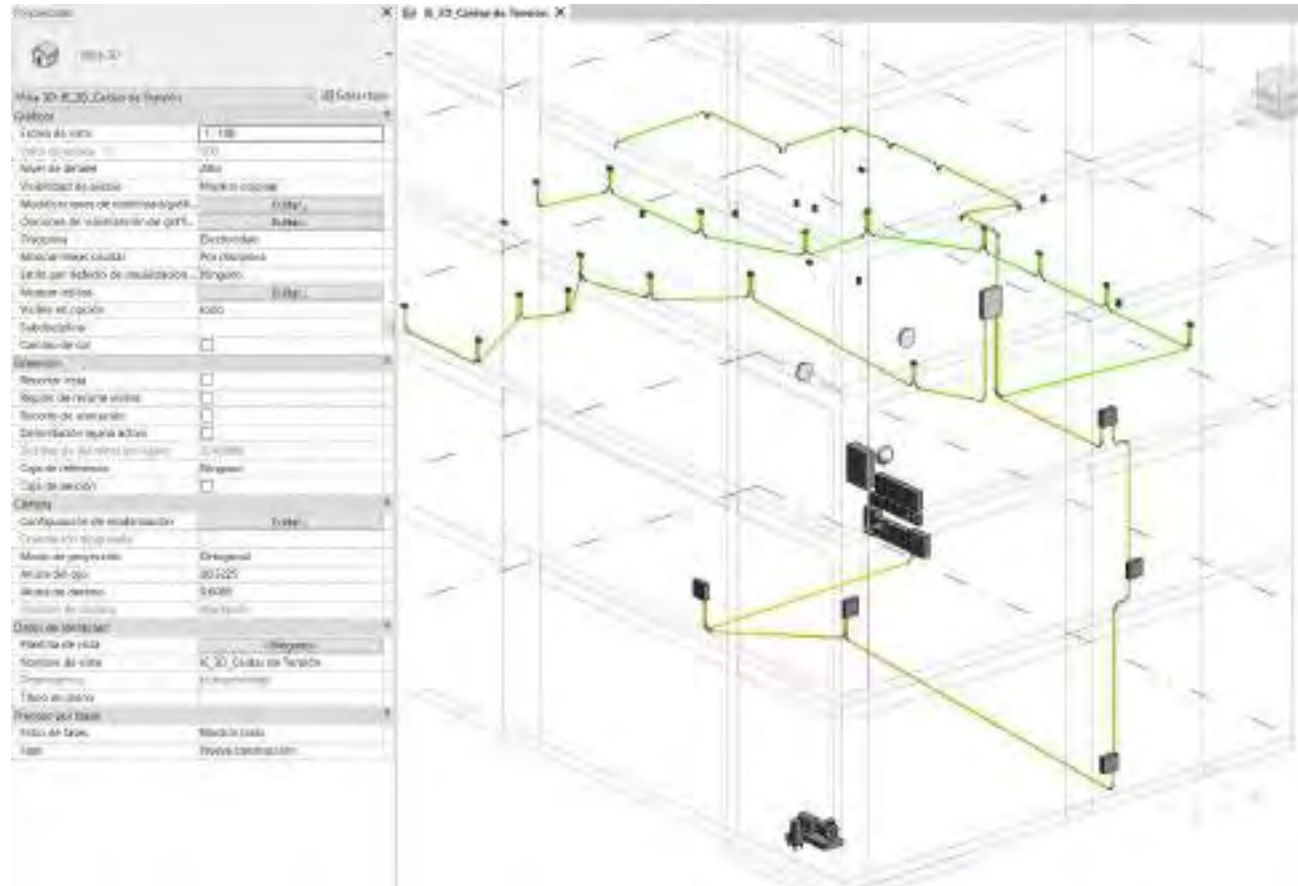
Número de la línea	Eje	Código de Cable	Longitud	Espec	Sección	RV
1	0-3	107	11.5422134	03824	4.0	0.111817407
2	0-1	210406803	11.5422134	00821	3.0	0.140241884
3	0-0	0.0000000	21.0844267	03824	4.0	0.111817407
4	0-1	210406803	21.0844267	00821	3.0	0.140241884

*Fuente:* Captura de software propia.



Figura 6.53

Vista tridimensional generada con rango de colores para interpretación de Caídas de Tensión



Fuente: Captura de software propia.

En caso de que no se hayan superado los rangos de color rojo en el resultado visual, esto indica que los niveles permitidos de caídas de tensión no han sido excedidos en el tramo resaltado.

En relación a los registros de tiempo, los mismos pueden ser consultados en las figuras 6.54 y 6.55. Los tiempos de ejecución detallados se presentan a continuación:

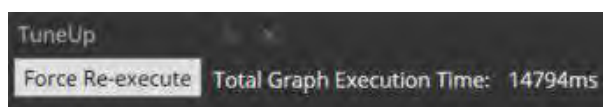
$$t_{ejecución} = 14\,794\text{ ms}$$

$$t_{Interfaz1} = 12\,815\text{ ms}$$

$$t_{Neto} = 1\,979\text{ ms}$$

### Figura 6.54

*Tiempo de ejecución del script “Caídas de Tensión”*



*Fuente:* Captura de software propia.

El diseño de proyectos implica una evaluación de posibles expansiones en la instalación, y el profesional a cargo, basado en su conocimiento e interpretación del proyecto, puede sugerir cambios en los calibres de los conductores. En este contexto, este script se adapta para funcionar con los calibres que el usuario proporcione.

Considerando la figura 6.52 previamente presentada, se puede destacar una gran diferencia en los resultados del script en dos posibles escenarios. En el caso anteriormente ejecutado, se planteó un calibre de alimentador de 6 mm<sup>2</sup>. Sin embargo, si se permitiera que Revit calcule automáticamente en función de la corriente soportable del material, que en este caso particular es una sección de 2,5 mm<sup>2</sup>, las caídas de tensión visualizadas difieren en gran medida, como se muestra en la figura Figura 6.56.

Aunque el resultado que se acerca al tono naranja podría interpretarse como una caída de tensión cercana al límite, numéricamente podría ser considerado aceptable. Sin embargo, al considerar el aspecto normativo, que también establece secciones mínimas para circuitos derivados y alimentadores, el resultado obtenido podría llevar a descartar por completo la propuesta de diseño debido a su inconformidad con la norma. Por esta razón, la incorporación de un parámetro de corrección en modelos BIM cobra importancia dentro de los objetivos de diseño, funcionamiento y conformidad normativa.

Figura 6.55

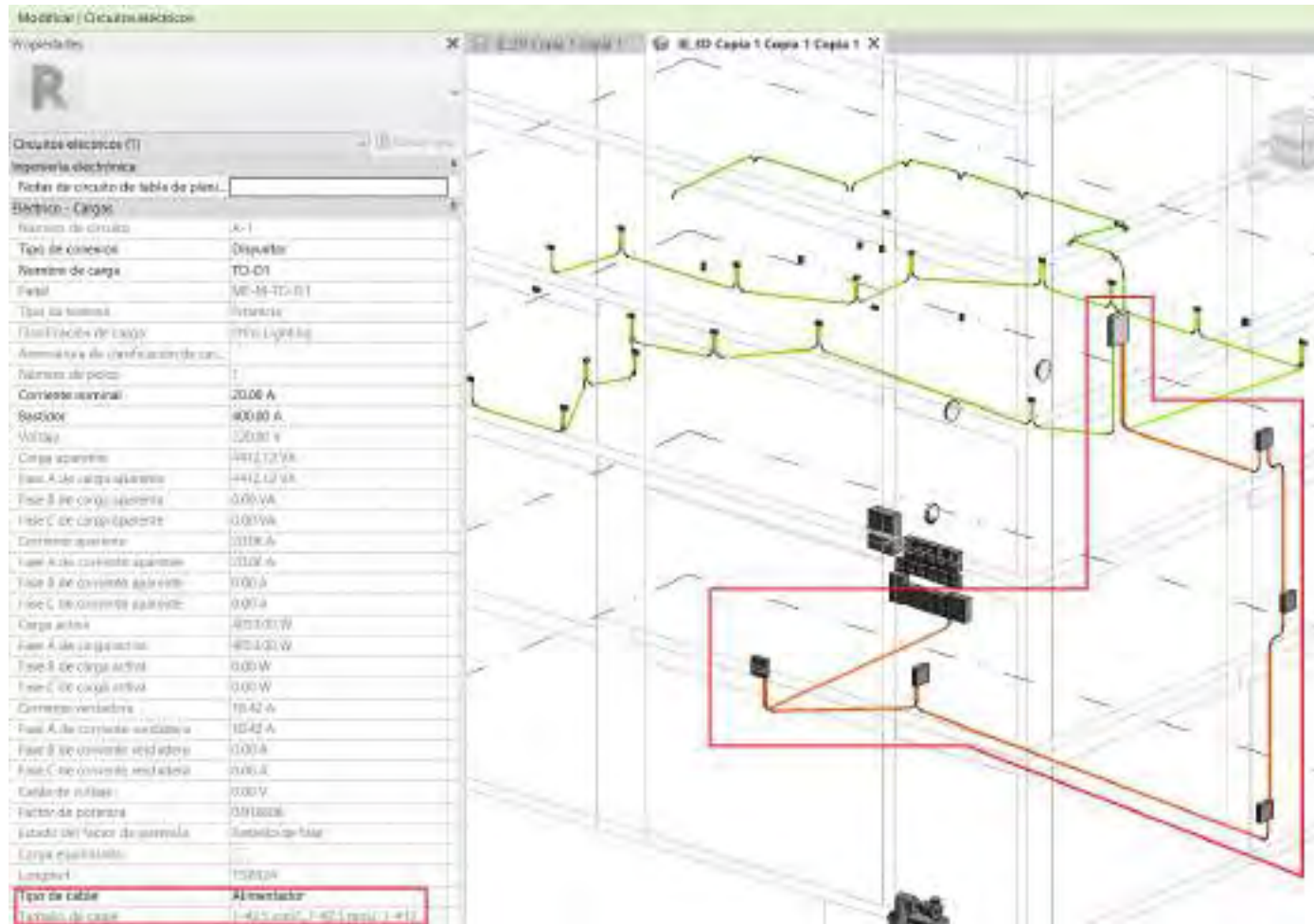
Tiempo de ejecución por nodo del script “Caídas de Tensión”

#	Name	Execution Time (ms)
36	Code Block	0
45	Code Block	0
21	Blockout	0
88	Math Sum	0
34	Code Block	0
52	List GetItemAtIndex	1
62	List Join	1
9	ScheduleView.SchedulableFields	1
80	Code Block	1
57	Code Block	1
10	Color Palette	1
18	Code Block	1
61	Code Block	1
19	List Names	1
35	Code Block	1
51	Element (Type /Name	1
32	Code Block	1
27	List Join	1
56	Code Block	1
29	Code Block	1
6	Categories	1
50	Element.GetParameterByName	2
55	Element.GetParameterByName	2
43	Code Length	2
50	Element.GetParameterByName	2
52	Element.GetParameterByName	2
53	Element.GetParameterByName	5
80	Element.GetParameterByName	6
11	---	8
8	ScheduleView.CreateSchedule	9
25	SelectMixedElements Data (order)	9
26	SelectMixedElements Data	10
6	Jobs (as List) by Param	11
48	Python Script	14
58	Python Script	16
47	Element.GetParameterByName	24
39	All List (by CategoryName	41
40	Element Location Curve	170
38	Elements in Connected Network	183
62	Element Over de ColorView	264
50	Key Schedule	1138
25	MultipleInPorts ++	12915
Execution Time Of Current Run		14794

Fuente: Captura de software propia.

**Figura 6.56**

*Recálculo de caídas de tensión con el calibre automático de Revit*



*Fuente:* Captura de software propia.

## 6.5 Resultados

### 6.5.1 Consideraciones

En este apartado de resultados se inicia planteando las consideraciones en base al proceso de desarrollo e implementación de resultados.

En ese sentido, primeramente la base programable sobre la cual ha derivado desarrollo la propuesta de la presente tesis ha sido el entorno llamado Dynamo.

Posteriormente, el software BIM sobre el que se ha aplicado la propuesta ha sido Autodesk Revit 2022 bajo una base arquitectónica modelada en su totalidad para un uso BIM en un mismo archivo para el desarrollo interdisciplinario.

A continuación se exponen los resultados obtenidos.

### 6.5.2 Tiempos de ejecución

Los tiempos expuestos en la Tabla 6.3, primeramente se deben entender como el resultado de un cálculo realizado en base a una diferencia entre el tiempo total de ejecución y el tiempo invertido en interfaces por acción manual de toma de datos. En ese sentido, los tiempos totales de ejecución fueron extraídos de la misma interfaz de Dynamo.

**Tabla 6.1**

*Resultados de tiempos de ejecución*

Tarea	Módulo	Tiempo de Ejecución
Diseño	Iluminación óptima	508 ms
	Conductos 1	713 ms
	Tomacorrientes	208 ms
	Conductos I	855 ms
	Conductos II	2191 ms
Cálculo	Cargas (Local Comercial)	2611 ms
	Cargas (Vivienda)	2869 ms
	Caídas de tensión	1979 ms

*Fuente:* Elaboración propia.

### 6.5.3 Concordancia BIM

Las dimensiones BIM han sido calificadas en base a la base teórica expuesta por Gonzales Villamil y Lesmes Fabián (2017) (Ver Tabla 2.1); esto claramente diferenciando las tareas dirigidas dentro del proceso de ejecución entre lo referente al diseño y cálculo.

**Tabla 6.2**

*Resultados BIM*

Tarea	Módulo	Dimensión BIM
Diseño	Iluminación óptima	3D
	Conductos 1	
	Tomacorrientes	
	Conductos I	
Cálculo	Conductos II	7D (Parcialmente)
	Cargas (Local Comercial)	
	Cargas (Vivienda)	
	Caídas de tensión	

*Fuente:* Elaboración propia.

### 6.5.4 Concordancia normativa

**Tabla 6.3**

*Resultados de concordancia normativa*

Tarea	Módulo	Base Normativa
Diseño	Iluminación óptima	Requisitos de iluminación RNE - EM010
	Conductos 1	CNE-U
	Tomacorrientes	
	Conductos I	
Conductos II		
Cálculo	Cargas (Local Comercial)	CNE-U
	Cargas (Vivienda)	
	Caídas de tensión	

*Fuente:* Elaboración propia.

En relación con las normas peruanas, se ha mantenido la base de diseño y representación de acuerdo a las normas de simbología establecidas por la DGE.

En particular, en el módulo dedicado a la iluminación, se ha cumplido con las especificaciones requeridas por el RNE, en su norma EM-010, referentes a los niveles mínimos de iluminación.

Para otros aspectos, se ha fundamentado el cumplimiento normativo en las pautas proporcionadas por el CNE-Utilización.

## 6.6 Discusión

En consideración a la problemática que se planteó al inicio de esta tesis, que señalaba la alta complejidad del proceso de modelado y cálculo de la especialidad de instalaciones eléctricas; siendo también exhibido en el tercer capítulo del presente trabajo un análisis que enseña que el desarrollo de estas tareas toman inversiones de tiempo en orden de minutos, horas e incluso días; los resultados revelan una reducción significativa de estos tiempos a una escala de milisegundos. Esto se ha logrado mediante la implementación de nodos programados en Dynamo.

Es importante destacar que los tiempos de ejecución en tareas de diseño y cálculo se obtuvieron directamente de la interfaz de Dynamo. Esta medición es relevante, ya que demuestra que los scripts se ejecutan de manera directa, sin requerir software adicional como intermediario. Esto optimiza el proceso de modelado.

A su vez, el modelo sobre el cual se aplica su ejecución tiene la característica de poseer la capacidad de modelar instalaciones eléctricas, gracias a la capacidad inherente del software y la forma en que canaliza la especialidad. Esto incluye la interrelación de especialidades, en este caso particular, entre la Arquitectura y las Instalaciones Eléctricas. Esta interconexión se logra a través de un vínculo en un solo archivo.

Además, los resultados reflejan cómo la implementación de los scripts ha afectado la capacidad del software BIM para abarcar dimensiones BIM. En el ámbito del modelado, se ha logrado representar la tercera dimensión (3D) de manera efectiva, lo que proporciona un nivel de detalle mucho mayor en comparación con las representaciones bidimensionales convencionales. Esto es fundamental para la comprensión detallada de los proyectos.

En lo que respecta al cálculo, la implementación de los scripts ha permitido ingresar al ámbito de la gestión BIM. Esto implica la capacidad adicional de realizar verificaciones de códigos normativos (*Code Checking*). En particular, se ha definido un cuadro de cargas según las normas peruanas, lo que es esencial para establecer la carga eléctrica de manera normativa.

Sin embargo, es importante mencionar que el cuadro de caídas de tensión, que también es relevante para la gestión y funcionamiento de la instalación eléctrica, se encuentra parcialmente relacionado con la séptima dimensión BIM. A pesar de estos avances, aún se requiere una evaluación más integral de otros aspectos del proyecto para completar la séptima dimensión.

Además, la propuesta ha logrado una independencia en el manejo de cálculo por defecto del software, que se comprobó que no se adecuaba a los requisitos específicos del contexto peruano. Esta adaptación exitosa a las normativas peruanas en un software extranjero es un logro importante de la investigación.

Se explica de esta forma que, con el uso de una herramienta paramétrica–visual para tareas de programación, se logran obtener resultados con enfoque BIM y cumplimiento normativo en tiempos mínimos; si esta herramienta se aplica directamente en un software con capacidad BIM con el que presente compatibilidad y tipo de operación directa en un solo archivo que brinde los alcances arquitectónicos.

Por último, los resultados no se limitan al ámbito interno del software, ya que se pueden exportar a vistas de documentación en planos. Aunque esta tarea requiere una interacción manual para definir los elementos visuales en los planos, la propuesta brinda diversas posibilidades de representación, incluida la representación tridimensional y la exposición de cálculos. Esto mejora la comprensión del proyecto en sus etapas posteriores.

Aunque los objetivos de esta tesis no incluyeron el desarrollo integral de un proyecto, en el Anexo D se incluyen láminas que muestran posibles distribuciones en distintos ámbitos representativos de planos dedicados a la especialidad de instalaciones eléctricas. Estos ejemplos ilustran el uso constante de representación tridimensional y la exposición de cálculos, lo que demuestra el potencial de la propuesta en proyectos de mayor envergadura.



## CONCLUSIONES

1. Se ha demostrado la posibilidad y capacidad de desarrollar propuestas informáticas basadas en programación paramétrica-visual. Específicamente, se ha logrado la creación de varios scripts que conforman una herramienta para el diseño y cálculo de instalaciones eléctricas interiores.
2. Durante la elaboración de la propuesta, se han descubierto las características inherentes de un software BIM. Esto ha llevado a definir tres aspectos clave: Necesidad, Información y Gestión. Estos elementos fundamentales constituyen la base de la metodología BIM aplicada a proyectos de instalaciones eléctricas interiores, validando así la hipótesis planteada.
3. Se han hallado posibilidades para crear programas informáticos destinados al campo de las instalaciones eléctricas interiores. Sin embargo, se ha identificado una limitación en cuanto a la compatibilidad de la interfaz de programación paramétrica-visual con el software BIM en el cual se pretende implementar.  

Esta limitante ha sido explorada en el contexto de utilizar la interfaz de *Dynamo* con un software BIM distinto a *Autodesk Revit*, confirmando así la hipótesis relacionada con la limitación de compatibilidad.
4. El desarrollo de la propuesta, en concordancia con los objetivos establecidos, ha seguido un enfoque algorítmico coherente. La ejecución de los scripts se ha basado en la interpretación de la normativa, consideraciones constructivas y las capacidades del software. Cada script ha sido diseñado con una entrada de datos específica y ha generado un resultado final, logrando así la convergencia de los scripts.
5. La aplicación ha permitido verificar funcionamiento de la propuesta en un modelo arquitectónico real. La propuesta ha sido capaz de resaltar los requerimientos de la normativa peruana referente al diseño eléctrico, así como las dimensiones BIM abarcadas en cada ejecución. Además, se ha confirmado la hipótesis de que los scripts tienen la capacidad de generar modelos BIM con facilidad, dado que los tiempos de ejecución han sido extremadamente reducidos (en el orden de segundos).

## RECOMENDACIONES

A partir del desarrollo del presente trabajo, se han identificado áreas de interés y consideraciones específicas que pueden enriquecer la formación profesional en Ingeniería Eléctrica y la aplicación de metodologías BIM en este campo y a los profesionales interesados en esta rama de estudio. Por lo tanto, se hacen las siguientes recomendaciones:

1. **Incorporación de la metodología BIM en la formación académica:** Se recomienda a la Escuela Profesional de Ingeniería Eléctrica de la Universidad Nacional de San Antonio Abad del Cusco, como también a otras instituciones de educación superior con programas relacionados a la Ingeniería Eléctrica, que integren contenidos introductorios de metodología BIM en cursos como Instalaciones Eléctricas o Dibujo Asistido por Computadora. Esta inclusión permitirá a los estudiantes familiarizarse con conceptos y herramientas BIM relevantes para su futura carrera.
2. **Fomentar la investigación en programación digital para ingeniería eléctrica:** Se insta a los departamentos académicos y a los profesionales interesados en el campo de la ingeniería eléctrica a promover la investigación en programación digital con un enfoque específico en temas no convencionales. Esta investigación puede abrir nuevas perspectivas y soluciones innovadoras dentro de la especialidad, impulsando el desarrollo de herramientas y aplicaciones especializadas.
3. **Evaluación crítica de software extranjero:** Se aconseja a las organizaciones y profesionales que consideren el uso de software de ingeniería desarrollado en el extranjero que realicen una evaluación exhaustiva de su adecuación a los requisitos normativos y descriptivos locales. En caso de incompatibilidades, se les recomienda considerar el desarrollo de herramientas informáticas propias que permitan la adaptación.
4. **Adopción de metodologías BIM en la industria de la construcción:** Se insta a las empresas del sector de la construcción, tanto en el Perú como en el extranjero, a adoptar metodologías BIM y a desarrollar flujos de trabajo eficientes. Esto contribuirá a mejorar la calidad de las edificaciones, beneficiando a la sociedad y fomentando el desarrollo de capacidades BIM a nivel nacional.
5. **Desarrollo integral de proyectos BIM:** Se sugiere a los profesionales proyectistas que los proyectos BIM no se limiten simplemente a la migración de instalaciones eléctricas desde soft-

ware CAD a entornos BIM. En su lugar, se recomienda que los proyectos BIM aborden aspectos de Necesidad, Información y Gestión, para así maximizar las ventajas de la metodología BIM.

6. **Fomentar la colaboración y mejora continua:** Se anima a los estudiantes y profesionales interesados en la programación digital y las instalaciones eléctricas a utilizar los scripts presentados en esta tesis como punto de partida. Se les insta a colaborar, mejorar y expandir estas herramientas con el propósito de contribuir al avance tecnológico y la optimización de procesos en el campo de la ingeniería eléctrica.

## BIBLIOGRAFÍA

- Almeida, A. (2019). BIM en el Perú. *Columnistas RPP*. [https://www.researchgate.net/profile/Alexandre-Del-Savio/publication/338412631\\_BIM\\_en\\_el\\_Peru/links/5e13adcaa6fdcc28375daf6d/BIM-en-el-Peru.pdf](https://www.researchgate.net/profile/Alexandre-Del-Savio/publication/338412631_BIM_en_el_Peru/links/5e13adcaa6fdcc28375daf6d/BIM-en-el-Peru.pdf)
- Autodesk Latinoamérica. (2022a). *AutoCAD Logo* [Logo]. <https://latinoamerica.autodesk.com/products/autocad/overview>
- Autodesk Latinoamérica. (2022b). *Navisworks Manage Logo* [Logo]. <https://latinoamerica.autodesk.com/products/navisworks/overview>
- Autodesk Latinoamérica. (2022c). *Revit Logo* [Logo]. <https://latinoamerica.autodesk.com/products/revit/overview>
- Bakhshi, S., Chenaghloou, M. R., Pour Rahimian, F., Edwards, D. J. y Dawood, N. (2022). Integrated BIM and DfMA parametric and algorithmic design based collaboration for supporting client engagement within offsite construction. *Automation in Construction*, 133, 104015. <https://doi.org/10.1016/j.autcon.2021.104015>
- Benavente Lepe, M. (2019). *Diseño de modelos BIM para infraestructuras civiles con DYNAMO* [Tesis de lic.]. Universidad de Sevilla. <https://idus.us.es/handle/11441/94058>
- Bianchini, C. y Nicastro, S. (2018). From BIM to H-BIM. *2018 3rd Digital Heritage International Congress (DigitalHERITAGE) held jointly with 2018 24th International Conference on Virtual Systems & Multimedia (VSMM 2018)*, 1-4. <https://doi.org/10.1109/DigitalHeritage.2018.8810087>
- Burnett, M. M. y McIntyre, D. W. (1995). Visual programming. *Los Alamitos*, 28, 11.
- CAD Outsourcing-Engineering Design Services. (2018). *Clash Detection BIM* [Modelo 3D]. <https://cadoutsourcingserviceses.blogspot.com/2018/10/bim-clash-detection-essential-for.html>
- Chavez Guerrero, R. y Vilchez Chung, S. (2021). *Propuesta de uso de la herramienta dynamo para optimizar el tiempo en el proceso de valorización de subcontratistas de acabados en edificios multifamiliares menores de 10 pisos de NSE A y B ejecutado por una Pyme constructora en Lima Metropolitana* [Tesis de Pregrado]. Universidad Peruana de Ciencias Aplicadas. <https://repositorioacademico.upc.edu.pe/handle/10757/656760>
- Chen, N., Kan, F., Wang, C., Wang, X., Qi, N., Liu, X., Mao, Y. y Wang, B. (2016). Application of BIM in Building Electrical Design. *Proceedings of the 2016 International Conference on Education, Management, Computer and Society*, 759-762. <https://doi.org/10.2991/emcs-16.2016.185>

- Contraloría General de la República del Perú. (2022). *Reporte de Obras Paralizadas 2022*. <https://www.gob.pe/institucion/contraloria/informes-publicaciones/3017117-presentacion-obras-y-proyectos-paralizados>
- Demchak, G., Dzambazova, T. y Krygiel, E. (2009). *Introducing Revit Architecture 2009* (1.ª ed.). Wiley.
- DIAL GmbH. (2022). *Dialux EVO 10.1 Logo* [Logo]. <https://www.dialux.com/en-GB/>
- Dirroots AEC Industry. (2022). *Dirroots One Logo* [Logo]. <https://dirroots.com>
- Dueñas Ponce de Leon, R. N. (2018). 1ra Unidad – Planificación y Elaboración de Proyectos [Plan de Estudios 2005]. En *IE552AEI: Instalaciones Eléctricas I*. Universidad Nacional de San Antonio Abad del Cusco.
- Dynamo BIM. (2022). *Dynamo Logo* [Logo]. <https://dynamobim.org/>
- Dynamo Primer. (2019). *¿Qué es la programación visual? | Dynamo Primer*. Consultado el 25 de febrero de 2022, desde [https://primer.dynamobim.org/es/01\\_Introduction/1-1\\_what\\_is\\_visual\\_programming.html](https://primer.dynamobim.org/es/01_Introduction/1-1_what_is_visual_programming.html)
- Egea, J., Liébana, Ó. y Gomez, M. (2015). Uso de BIM como herramienta de integración en talleres de Tecnología de la Edificación. <http://hdl.handle.net/11268/5136>
- Fuentes-Rosado, J. I. y Moo-Medina, M. . (2017). Dificultades de aprender a programar. *Revista Educación en Ingeniería*, 12(24), 76-82. <https://doi.org/10.26507/rei.v12n24.728>
- Gallardo, L. y Guadalupe, J. (2017). *Salto al BIM: Estrategias BIM de calidad para empresas punteras del sector AEC*. J.H. Guadalupe.
- GeeksforGeeks. (2022). Python Program for Fibonacci numbers. <https://www.geeksforgeeks.org/python-program-for-program-for-fibonacci-numbers-2>
- Gonzales Villamil, W. R. y Lesmes Fabián, C. A. (2017). Siete dimensiones de un proyecto de construcción con la metodología Building Information Modeling. *L'esprit Ingénieur*, 8(1). <http://revistas.ustatunja.edu.co/index.php/lingenieur/article/view/1659>
- González Marquez, R. J., Gámez, F. y Severino, M. (2014). INTRODUCCION A LA METODOLOGÍA BIM. *The Spanish Journal of BIM*, 14, 48-54.
- Grasshopper 3D. (2022). *Grasshopper 3D algorithmic modeling Logo* [Logo]. <https://www.grasshopper3d.com/>
- Kensek, K. M. (2014). Integration of Environmental Sensors with BIM: case studies using Arduino, Dynamo, and the Revit API. *Informes de la Construcción*, 66(536), 3-3. <https://doi.org/10.3989/ic.13.151>
- Kirschner, M. J. (2015). *Visual programming in three dimensions : visual representations of computational mental models* [Tesis de Maestría]. Massachusetts Institute of Technology. Department of Architecture. <https://dspace.mit.edu/handle/1721.1/99246>

- Ley 23560 – Sistema Legal de Unidades de Medida del Perú. (1982). Diario Oficial el Peruano, 6 de enero de 1983. <https://docs.peru.justia.com/federales/leyes/23560-dec-31-1982.pdf>
- MacLeamy, P. (2008). BIM, BAM, BOOM! How to build greener, high-performance buildings. *Urban Land Green Magazine*.
- Madrid, J. A. (2014). Nivel de desarrollo LOD. Definiciones, innovaciones y adaptación a España. *Spanish Journal of Building Information Modelling*, 15(1), 40-56. <https://dialnet.unirioja.es/servlet/articulo?codigo=5496892>
- McConnell, S. (2004). *Code Complete* (2.<sup>a</sup> ed.). Pearson Education.
- Mcfarland, J. E. (2007). *Building Information Modelling for MEP* (inf. téc.). Department of Architectural Engineering y Construction Science, Kansas State University. Manhattan, Kansas. <https://krex.k-state.edu/dspace/bitstream/handle/2097/518/jessicamcfarland2007.pdf?sequence=1>
- Ministerio de Economía y Finanzas. (2021). *Guía Nacional BIM - Gestión de la información para inversiones desarrolladas con BIM* (inf. téc.). [https://www.mef.gob.pe/planbimperu/docs/recursos/guia\\_nacional\\_BIM.pdf](https://www.mef.gob.pe/planbimperu/docs/recursos/guia_nacional_BIM.pdf)
- Ministerio de Energía y Minas. (2006). *Código Nacional de Electricidad – Utilización*. Dirección General de Electricidad. Lima – Peru.
- Ministerio de Energía y Minas. (2008). *Manual de Sustentación del Código Nacional de Electricidad– Utilización 2006*. Dirección General de Electricidad.
- Ministerio de Vivienda, Construcción y Saneamiento. (2006). *Reglamento Nacional de Edificaciones*. Dirección General de Políticas y Regulación en Construcción y Saneamiento. Lima – Peru.
- Monteiro, A. (2016). Visual programming language for creating BIM models with level of development 400. *Proc., 4th BIM Int. Conf. São Paulo, Brazil: BIMMI*.
- Murguía Sánchez, D., Vasquez, C., Balboa, M. y Lara, W. (2021). Segundo Estudio de Adopción BIM en Proyectos de Edificación en Lima y Callao. Pontificia Universidad Católica del Perú. Departamento de Ingeniería. <http://repositorio.pucp.edu.pe/index/handle/123456789/176216>
- Nallaperumal, K. y Krishnan, A. (2013). *Engineering Research Methodology A Computer Science and Engineering and Information and Communication Technologies Perspective*.
- Nezamaldin, D. (2019). *Parametric design with Visual programming in Dynamo with Revit* [Tesis de maestría, KTH Royal Institute of Technology in Stockholm]. <https://www.diva-portal.org/smash/get/diva2:1278919/FULLTEXT01.pdf>
- Noa Mayta, A. y Vergara Ovalle, O. (2021). *Diseño de las Instalaciones Eléctricas, Especiales aplicando tecnología BIM, Automatización y Autogeneración Híbrida, de un Hotel 5 Estrellas* [Tesis

- de Pregrado]. Universidad Nacional de San Antonio Abad del Cusco. <http://repositorio.unsaac.edu.pe/handle/20.500.12918/6414>
- Noves, J. (2015). *Lenguajes de programación visual en el mundo BIM* (Ponencia). 2do Congreso Lationamericano “BIM Latam 2015”. Santiago, Chile. <https://docplayer.es/11427562-Lenguajes-de-programacion-visual-en-el-mundo-bim-valentin-noves.html>
- Oh, M., Lee, J., Hong, S. W. y Jeong, Y. (2015). Integrated system for BIM-based collaborative design. *Automation in Construction*, 58, 196-206. <https://doi.org/10.1016/j.autcon.2015.07.015>
- Popov, L. (2016). *Implementation of BIM in construction project*. [Tesis de Bachillerato]. Saimaa University of Applied Sciences. <https://www.theseus.fi/bitstream/handle/10024/120667/Implementation+of+BIM++Thesis++Leonid+Popov+DDCiv16.pdf?sequence=1>
- Python Software Foundation. (2022). *Python Logo* [Logo]. <https://www.python.org/>
- Rojas Blas, W. N. (2021). *Propuesta de workflow BIM/VDC para el diseño geométrico de vías de transitabilidad de la Zona Q - Huaycán Ate-Lima* [Tesis de Pregrado]. Universidad Cesar Vallejo. <https://repositorio.ucv.edu.pe/handle/20.500.12692/71151>
- Sancha Rubio, D. (2022). *Análisis comparativo de herramientas y procesos BIM para diseño y cálculo de instalaciones en edificación*. [Trabajo de Fin de Grado]. Universidad de Valladolid. <https://uvadoc.uva.es/bitstream/handle/10324/54221/TFG-I-2254.pdf?sequence=1>
- Shu, N. C. (1986). Visual Programming Languages: A Perspective and a Dimensional Analysis. *Visual Languages*, 11-34. [https://doi.org/10.1007/978-1-4613-1805-7\\_2](https://doi.org/10.1007/978-1-4613-1805-7_2)
- Tharwat, A. C. (2010). *LISP Block Coordinates*. <https://autolispprograms.wordpress.com/>
- Trebbi, C., Cianciulli, M., Matarazzo, F., Mirarchi, C., Cianciulli, G. y Pavan, A. (2019). Clash Detection and Code Checking BIM Platform for the Italian Market. *Digital Transformation of the Design, Construction and Management Processes of the Built Environment*, 115-125. [https://doi.org/10.1007/978-3-030-33570-0\\_11](https://doi.org/10.1007/978-3-030-33570-0_11)
- Vizcarra Aparicio, J. M. (2018). *Reporte Regional del Building Information Modeling (BIM) Cusco 2018*. Consultado el 12 de febrero de 2022, desde [https://www.academia.edu/37244259/Reporte\\_Regional\\_del\\_Building\\_Information\\_Modeling\\_BIM\\_Cusco\\_2018](https://www.academia.edu/37244259/Reporte_Regional_del_Building_Information_Modeling_BIM_Cusco_2018)
- XAL Company. (2022). *Building Information Modeling - The digital twin* [Ilustración]. <https://www.xal.com/en/revit-bim/>
- Xiaolei, W. (2018). Research on the Application of BIM Concept and BIM Software in Architectural Design. *2018 International Conference on Engineering Simulation and Intelligent Control (ESAIC)*, 218-220. <https://doi.org/10.1109/ESAIC.2018.00058>

## ANEXOS

## Anexo A: Código fuente desarrollado en Dynamo vía Python

A continuación se muestran los programas en forma de código fuente en lenguaje Python para su integración en la interfaz de Dynamo, estos mismos son la implementación de los algoritmos desarrollados en la presente tesis.

Código A.1: Implementación de los Algoritmos 1 y 2

```

1  # Recursos de Revit
2  import sys
3  import clr
4  clr.AddReference('ProtoGeometry')
5  from Autodesk.DesignScript.Geometry import *
6  clr.AddReference('RevitNodes')
7  import Revit
8  clr.ImportExtensions(Revit.Elements)
9  clr.ImportExtensions(Revit.GeometryConversion)
10 # Toma de datos desde Dynamo
11 Lev=IN[0]
12 TamEle=len(Lev)
13 Elev=IN[1]
14 Sup=IN[2]
15 Inf=IN[3]
16 ## Coordenadas Superiores e Inferiores
17 Xsup=IN[4]
18 Ysup=IN[5]
19 Zsup=IN[6]
20
21 Xinf=IN[7]
22 Yinf=IN[8]
23 Zinf=IN[9]
24 # Listas auxiliares para las lineas generadas y conectores utilizados
25 LIN=[0 for x in range(TamEle)]
26 USED=[0 for x in range(TamEle)]
27 #Inicializadores
28 Xo=0
29 Yo=0
30 Zo=0
31
32 Xff=0
33 Yff=0
34 Zff=0
35 #Implementacion del algoritmo 2
36 def Lin2(a,da,b,db):
37     Min=10000 #Hipotetica distancia inicial
38     if da==1:
39         Pi=Sup[a]
40         h=len(Sup[a])
41         Xo=Xsup[a]
42         Yo=Ysup[a]
43         Zo=Zsup[a]
44

```



```

45 elif da==-1:
46     Pi=Inf[a]
47     h=len(Inf[a])
48     Xo=Xinf[a]
49     Yo=Yinf[a]
50     Zo=Zinf[a]
51
52 if db==1:
53     Pf=Sup[b]
54     w=len(Sup[b])
55     Xff=Xsup[b]
56     Yff=Ysup[b]
57     Zff=Zsup[b]
58
59 elif db==-1:
60     Pf=Inf[b]
61     w=len(Inf[b])
62     Xff=Xinf[b]
63     Yff=Yinf[b]
64     Zff=Zinf[b]
65
66 M=[] # Lista vacia auxiliar
67
68 for m in range(len(Pi)):
69     if USED[a]!=0:
70         # Se establece que una aproximacion entre el conector ya usado con el punto
71         # inicial que se esta evaluando
72         if Geometry.IsAlmostEqualTo( USED[a] , Pi[m] ) is True:
73             M.append(Pi[m])
74             del Xo[m]
75             del Yo[m]
76             del Zo[m]
77
78 for m in range(len(M)):
79     Pi.remove(M[m]) #Se quita puntos iniciales ya usados
80
81 line=0 # Inicializador
82
83 for i in range(len(Pi)):
84     for j in range(len(Pf)):
85         #Toma de coordenadas XYZ iniciales y finales
86         Xi=Xo[i]
87         Yi=Yo[i]
88         Zi=Zo[i]
89
90         Xf=Xff[j]
91         Yf=Yff[j]
92         Zf=Zff[j]
93         #Calculo de la distancia entre dos puntos
94         DIS=( (Xf-Xi)**2 + (Yf-Yi)**2 + (Zf-Zi)**2 )**(0.5)
95         # Evaluacion con la distancia minima
96         if DIS<Min:
97             line=Line.ByStartPointEndPoint(Pi[i],Pf[j])
98             USED[b]=Pf[j]
99             # Actualizacion de la distancia minima
100            Min=DIS
101        else:
102            continue
103
104 return line
105 #Llamados a la funcion LIN2 del Algoritmo 2
106 for i in range(TamEle-1):
107     if Lev[i]==Lev[i+1]:
108         LIN[i]=Lin2(i,-1,i+1,-1)

```

```
107     else:
108         LIN[i]=Lin2(i,1,i+1,-1)
109
110 R=LIN
111 OUT = [R]
```

## Código A.2: Implementación de los Algoritmos 3 y 4

```

1 # Recursos de Revit
2 import sys
3 import clr
4 clr.AddReference('ProtoGeometry')
5 from Autodesk.DesignScript.Geometry import *
6 clr.AddReference('RevitNodes')
7 import Revit
8 clr.ImportExtensions(Revit.Elements)
9 clr.ImportExtensions(Revit.GeometryConversion)
10 # Toma de datos
11 Zi=IN[0]
12 Zf=IN[1]
13 Lev=IN[2]
14 ELev=IN[3]
15 # Inicializadores
16 Zna=0
17 Znb=0
18 R=[]
19 # Generado de lista de ceros
20 Tam=len(Lev)
21 ZN=[[0 for x in range(2)] for y in range(Tam-1)]
22
23 # Aplicacion del Algoritmo 4
24 def TransInf(a):
25     Zna=-Zi[a]+ELev[a]
26     Znb=-Zf[a]+ELev[a]
27     ZN[a]=[Zna,Znb]
28
29 def TransSup(a,b):
30     if ELev[a]<ELev[b]:
31         Zna=ELev[b]-Zi[a]
32         Znb=ELev[b]-Zf[a]
33         ZN[a]=[Zna,Znb]
34     else:
35         Zna=ELev[a]-Zi[a]
36         Znb=ELev[a]-Zf[a]
37         ZN[a]=[Zna,Znb]
38 # Calculo de movimiento en eje Z
39 for i in range(Tam-1):
40     if Lev[i]==Lev[i+1]:
41         TransInf(i)
42     else:
43         TransSup(i,i+1)
44
45 OUT = ZN

```

## Código A.3: Implementación de los Algoritmos 5 y 6

```

1 # Recursos de Revit
2 import sys
3 import clr
4 clr.AddReference('ProtoGeometry')
5 from Autodesk.DesignScript.Geometry import *
6 clr.AddReference('RevitNodes')
7 import Revit
8 clr.ImportExtensions(Revit.Elements)
9 clr.ImportExtensions(Revit.GeometryConversion)
10 # Toma de datos desde Dynamo
11 Lev=IN[0]
12 TamEle=len(Lev)
13 Sup=IN[1]
14 # Coordenadas Superiores
15 Xsup=IN[4]
16 Ysup=IN[5]
17 Zsup=IN[6]
18
19 # Listas auxiliares para las lineas generadas y conectores utilizados
20 LIN=[0 for x in range(TamEle)]
21 USED=[0 for x in range(TamEle)]
22 #Inicializadores
23 Xo=0
24 Yo=0
25 Zo=0
26
27 Xff=0
28 Yff=0
29 Zff=0
30 #Implementacion del algoritmo 6
31 def Lin2_I(a,b):
32     Min=10000 #Hipotetica distancia inicial
33     Pi=Sup[a]
34
35     h=len(Sup[a])
36     Xo=Xsup[a]
37     Yo=Ysup[a]
38     Zo=Zsup[a]
39
40     Pf=Sup[b]
41     w=len(Sup[b])
42     Xff=Xsup[b]
43     Yff=Ysup[b]
44     Zff=Zsup[b]
45
46     M=[] # Lista vacia auxiliar
47
48     for m in range(len(Pi)):
49         if USED[a]!=0:
50             # Se establece que una aproximacion entre el conector ya usado con el punto
51             # inicial que se esta evaluando
52             if Geometry.IsAlmostEqual( USED[a] , Pi[m] ) is True:
53                 M.append(Pi[m])
54                 del Xo[m]
55                 del Yo[m]
56                 del Zo[m]
57
58     for m in range(len(M)):
59         Pi.remove(M[m]) #Se quita puntos iniciales ya usados
60
61     line=0 # Inicializador

```

```

62     for i in range(len(Pi)):
63         for j in range(len(Pf)):
64             #Toma de coordenadas XYZ iniciales y finales
65             Xi=Xo[i]
66             Yi=Yo[i]
67             Zi=Zo[i]
68
69             Xf=Xff[j]
70             Yf=Yff[j]
71             Zf=Zff[j]
72             #Calculo de la distancia entre dos puntos
73             DIS=( (Xf-Xi)**2 + (Yf-Yi)**2 + (Zf-Zi)**2 )**(0.5)
74             # Evaluacion con la distancia minima
75             if DIS<Min:
76                 line=Line.ByStartPointEndPoint(Pi[i],Pf[j])
77                 USED[b]=Pf[j]
78                 # Actualizacion de la distancia minima
79                 Min=DIS
80             else:
81                 continue
82     return line
83 #Llamados a la funcion LIN2_I del Algoritmo 6
84 for i in range(TamEle-1):
85     LIN[i]=Lin2_I(i,i+1)
86
87 R=LIN
88 OUT = [R]

```

## Código A.4: Implementación de los Algoritmos 7 y 8

```

1  # Recursos de Revit
2  import sys
3  import clr
4  clr.AddReference('ProtoGeometry')
5  from Autodesk.DesignScript.Geometry import *
6  clr.AddReference('RevitNodes')
7  import Revit
8  clr.ImportExtensions(Revit.Elements)
9  clr.ImportExtensions(Revit.GeometryConversion)
10 # Toma de datos desde Dynamo
11 Lev=IN[0]
12 TamEle=len(Lev)
13 Inf=IN[1]
14 # Coordenadas Inferiores
15 Xinf=IN[2]
16 Yinf=IN[3]
17 Zinf=IN[4]
18
19 # Listas auxiliares para las lineas generadas y conectores utilizados
20 LIN=[0 for x in range(TamEle)]
21 USED=[0 for x in range(TamEle)]
22 #Inicializadores
23 Xo=0
24 Yo=0
25 Zo=0
26
27 Xff=0
28 Yff=0
29 Zff=0
30 #Implementacion del algoritmo 8
31 def Lin2_T(a,b):
32     Min=10000 #Hipotetica distancia inicial
33     Pi=Inf[a]
34
35     h=len(Inf[a])
36     Xo=Xinf[a]
37     Yo=Yinf[a]
38     Zo=Zinf[a]
39
40     Pf=Inf[b]
41     w=len(Inf[b])
42     Xff=Xinf[b]
43     Yff=Yinf[b]
44     Zff=Zinf[b]
45
46     M=[] # Lista vacia auxiliar
47
48     for m in range(len(Pi)):
49         if USED[a]!=0:
50             # Se establece que una aproximacion entre el conector ya usado con el punto inicial
51             # que se esta evaluando
52             if Geometry.IsAlmostEqual( USED[a] , Pi[m] ) is True:
53                 M.append(Pi[m])
54                 del Xo[m]
55                 del Yo[m]
56                 del Zo[m]
57
58     for m in range(len(M)):
59         Pi.remove(M[m]) #Se quita puntos iniciales ya usados
60
61     line=0 # Inicializador

```

```

62     for i in range(len(Pi)):
63         for j in range(len(Pf)):
64             #Toma de coordenadas XYZ iniciales y finales
65             Xi=Xo[i]
66             Yi=Yo[i]
67             Zi=Zo[i]
68
69             Xf=Xff[j]
70             Yf=Yff[j]
71             Zf=Zff[j]
72             #Calculo de la distancia entre dos puntos
73             DIS=( (Xf-Xi)**2 + (Yf-Yi)**2 + (Zf-Zi)**2 )**(0.5)
74             # Evaluacion con la distancia minima
75             if DIS<Min:
76                 line=Line.ByStartPointEndPoint(Pi[i],Pf[j])
77                 USED[b]=Pf[j]
78                 # Actualizacion de la distancia minima
79                 Min=DIS
80             else:
81                 continue
82     return line
83 #Llamados a la funcion LIN2_T del Algoritmo 8
84 for i in range(TamEle-1):
85     LIN[i]=Lin2_T(i,i+1)
86
87 R=LIN
88 OUT = [R]

```

## Código A.5: Implementación del Algoritmo 9

```
1 # Recursos de Revit
2 import sys
3 import clr
4 clr.AddReference('ProtoGeometry')
5 from Autodesk.DesignScript.Geometry import *
6 import math
7
8 # Toma de Area
9 A=IN[5]
10 # Parametros de calculo
11 PrimA=IN[2]
12 SegA=IN[3]
13 FracA=IN[4]
14
15 PrimP=IN[0]
16 SegP=IN[1]
17 FracP=IN[1]
18 # Inicializador
19 Pot=[0,0,0]
20 # Generacion de calculo
21 for i in range(3):
22     if i==0:
23         A=A-PrimA
24         Pot[i]=PrimP
25     elif i==1 and A>0:
26         A=A-SegA
27         Pot[i]=SegP
28     else:
29         while A>0:
30             A=A-FracA
31             Pot[i]=Pot[i]+FracP
32
33 OUT = Pot
```



## Código A.6: Implementación del Algoritmo 11

```
1 # Recursos de Revit
2 import sys
3 import clr
4 clr.AddReference('ProtoGeometry')
5 from Autodesk.DesignScript.Geometry import *
6
7 # Presencia de Cocina Electrica
8 NoCocEle = IN[0]
9 # Potencia Adicional
10 Pot_Ad=IN[1]
11 #Generacion de lista de recalcu lo por cocina electrica
12 Tam=len(Pot_Ad)
13 N_Pot=[0 for x in range (Tam)]
14 # Calculo indivual por cocina electrica
15 for i in range (Tam):
16     if NoCocEle!=0:
17         if Pot_Ad[i]>6000:
18             N_Pot[i]=Pot_Ad[i]+(Pot_Ad[i]-6000)*0.25
19         else:
20             N_Pot[i]=Pot_Ad[i]
21     else:
22         N_Pot[i]=Pot_Ad[i]*0.25
23
24 OUT=N_Pot
```

## Código A.7: Implementación del Algoritmo 10

```

1 # Recursos de Revit
2 import sys
3 import clr
4 clr.AddReference('ProtoGeometry')
5 from Autodesk.DesignScript.Geometry import *
6
7 # Datos de Entrada
8 PotAd=IN[0] # Potencia Adicional
9 NombAd=IN[1] # Nombre de la Carga Adicional
10 Norm=IN[2] # Norma de Referencia
11 MDC=IN[3] # Maxima Demanda
12 #Generacion del orden vertical de la matriz de cargas adicionales
13 TamVert=len(PotAd)
14 M=[[" for i in range(6) for j in range(TamVert+1)]
15 # Inicializadores
16 PI=0
17 MD=0
18 # Calculo y ubicacion
19 for i in range(TamVert+1):
20     for j in range(6):
21         if i==0 and j==0:
22             M[i][j]=Norm
23         if j==1 and i==0:
24             M[i][j]="Carga Adicional"
25         if j==1 and i!=0:
26             M[i][j]=NombAd[i-1]
27         if j==2:
28             M[i][j]=" "
29         if j==3 and i>0:
30             M[i][j]=PotAd[i-1]
31             PI=PI+PotAd[i-1]
32         if j==4 and i>0:
33             if PotAd[i-1]==MDC[i-1]:
34                 M[i][j]=1
35             else:
36                 M[i][j]=MDC[i-1]/PotAd[i-1]
37         if j==5 and i>0:
38             M[i][j]=MDC[i-1]
39             MD=MD+MDC[i-1]
40 # Totalizador
41 Tot=["", "", "Total", PI, "", MD]]
42 M.extend(Tot)
43
44 OUT = M

```

## Código A.8: Implementación del Algoritmo 12

```

1 # Recursos de Revit
2 import sys
3 import clr
4 clr.AddReference('ProtoGeometry')
5 from Autodesk.DesignScript.Geometry import *
6 # Datos de Entrada
7 Cir=IN[0] # Circuito
8 V=IN[1] # Tension de Operacion
9 NP=IN[2] # Nro de polos
10 Long=IN[3] # Longitud del tramo
11 Mat=IN[4] # Material del conductor
12 I=IN[5] # Corriente
13 S=IN[6] # Seccion del conductor
14 # Generacion de matriz de caidas de tension
15 tam=len(Cir)
16 M=[["Circuito", "I", "L", "Mat", "Seccion", "DV"] for i in range(tam)]
17 # Aplicacion de factores por conexion y material
18 for i in range(tam):
19     if NP[i]==1:
20         k=2
21     elif NP[i]==3:
22         k=3**(1/2)
23     if Mat[i]=="Cobre":
24         p=0.0176
25     elif Mat[i]=="Aluminio":
26         p=0.0260
27     #Calculo de la caida de tension
28     DV=k*I[i]*1.25*(Long[i]/S[i])*0.9*p
29     # Ubicacion de datos
30     M[i][0]=Cir[i]
31     M[i][1]=I[i]*1.25
32     M[i][2]=Long[i]
33     M[i][3]=Mat[i]
34     M[i][4]=S[i]
35     M[i][5]=DV
36
37 OUT = M

```

## Anexo B: Recursos código fuente de terceros

En la presente tesis se han utilizado recursos programados por terceros, a continuación se detallan los códigos fuentes implementados, detallando a su respectivo autor.

Código A.9: Herramienta de Tharwat (2010)

```

1 (defun c:THex (/ dir fNme ss)
2   ; THARWAT 2010
3   (if (and (setq dir (getvar 'dwgprefix))
4           (setq fNme (open (strcat dir "Block-coordinates.txt" ) "w"))
5           (setq ss (ssget "_.:L" '((0 . "INSERT"))))
6           )
7     (
8       (lambda (i / ssl e pt1 )
9         (while
10          (setq ssl
11            (ssname ss (setq i (1+ i))))
12          (setq e
13            (entget ssl))
14          (setq pt1
15            (cdr (assoc 10 e)))
16
17          (write-line
18            (strcat (rtos (car pt1) 2)
19                  "," (rtos (cadr pt1) 2)
20                  "," (rtos (caddr pt1) 2))
21            fNme)
22          )
23        )
24      -1
25    )
26 (princ)
27 )
28 (close fNme)
29 (princ "\n Written by Tharwat")
30 (princ)
31 )

```

**Anexo C: Autorización de uso de base arquitectónica**

La presente tesis, para lograr su objetivo utilizó una base arquitectónica diseñada y aprobada por un Arquitecto Proyectista Habilitado y Colegiado. A continuación se presenta la debida autorización de uso.

Cusco, 20 de octubre del 2022

### Carta de Autorización

La que suscribe, Silvia Luciana Jordan Ramos, Arquitecta con CAP N° 014989.

Autoriza:

Al Br. Luis Giuseppe Yiera Altamirano, autor de la tesis titulada:

***"Programación Paramétrica-Visual aplicada a Software BIM para el desarrollo de la Etapa de Diseño y Cálculo de Instalaciones Eléctricas Interiores"***

presentada a la Escuela Profesional de Ingeniería Eléctrica, de la Facultad de Ingeniería Eléctrica, Electrónica, Informática y Mecánica de la Universidad Nacional de San Antonio Abad del Cusco.

A hacer uso de la base arquitectónica denominada:

***"Edificio Comercial-Residencial Unión"***

material profesional el cual mi persona declara en AUTORÍA TOTAL.

Del mismo modo, la presente autorización limita el uso del material a solo y exclusivamente fines académicos, debiéndose dar los créditos correspondientes.

Se expide el presente documento a solicitud del interesado para continuar el desarrollo de su trabajo de tesis y eventuales trámites que se vieran por convenientes.



  
S. Luciana Jordan Ramos  
ARQUITECTA  
CAP. 14989

Silvia Luciana Jordan Ramos  
Arquitecta Proyectista  
CAP 014989

De Autoriza:

... (faint text)

**Anexo D: Planos**

En las siguientes láminas se exponen algunos planos que fueron resultado de la ejecución de los módulos propuesta sobre el entorno del software BIM Revit.



UNIVERSIDAD  
NACIONAL DE  
SAN ANTONIO  
ABAD DEL  
CUSCO

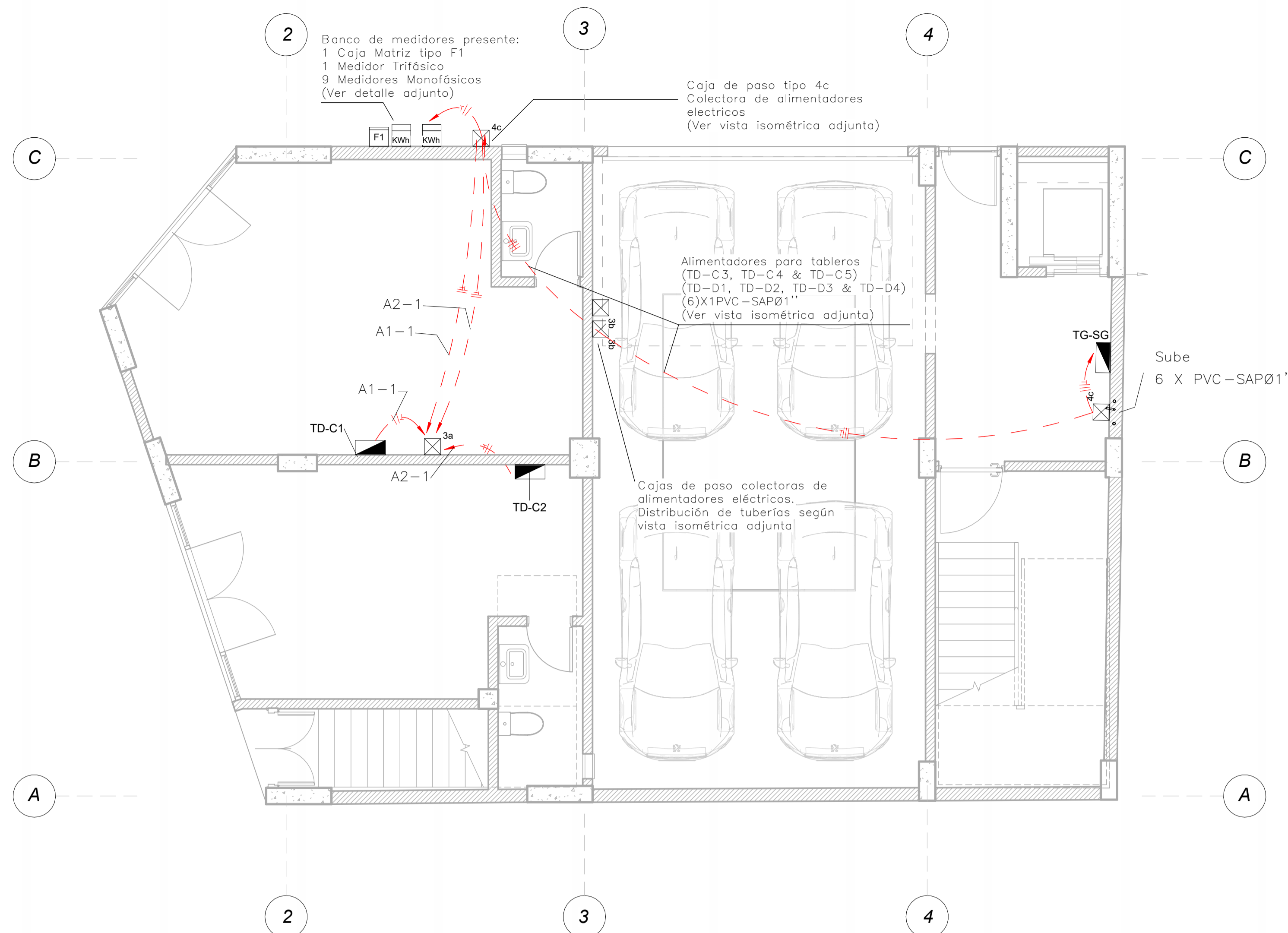
FACULTAD DE  
INGENIERÍA  
ELÉCTRICA,  
ELECTRÓNICA,  
INFORMÁTICA Y  
MECÁNICA

ESCUELA  
PROFESIONAL DE  
INGENIERÍA  
ELÉCTRICA

Alimentadores  
Eléctricos  
1er Nivel

LÁMINA:

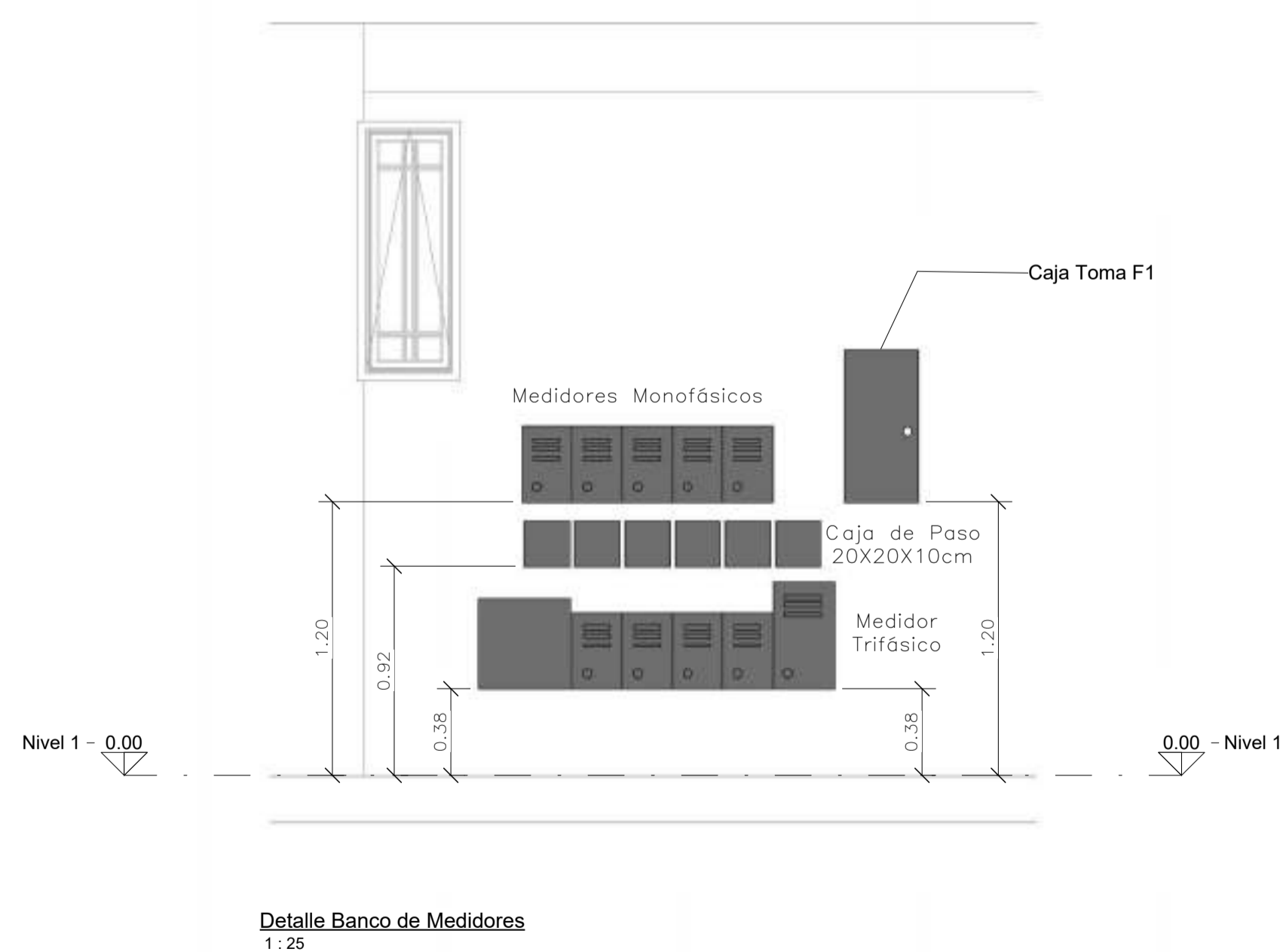
IE-01



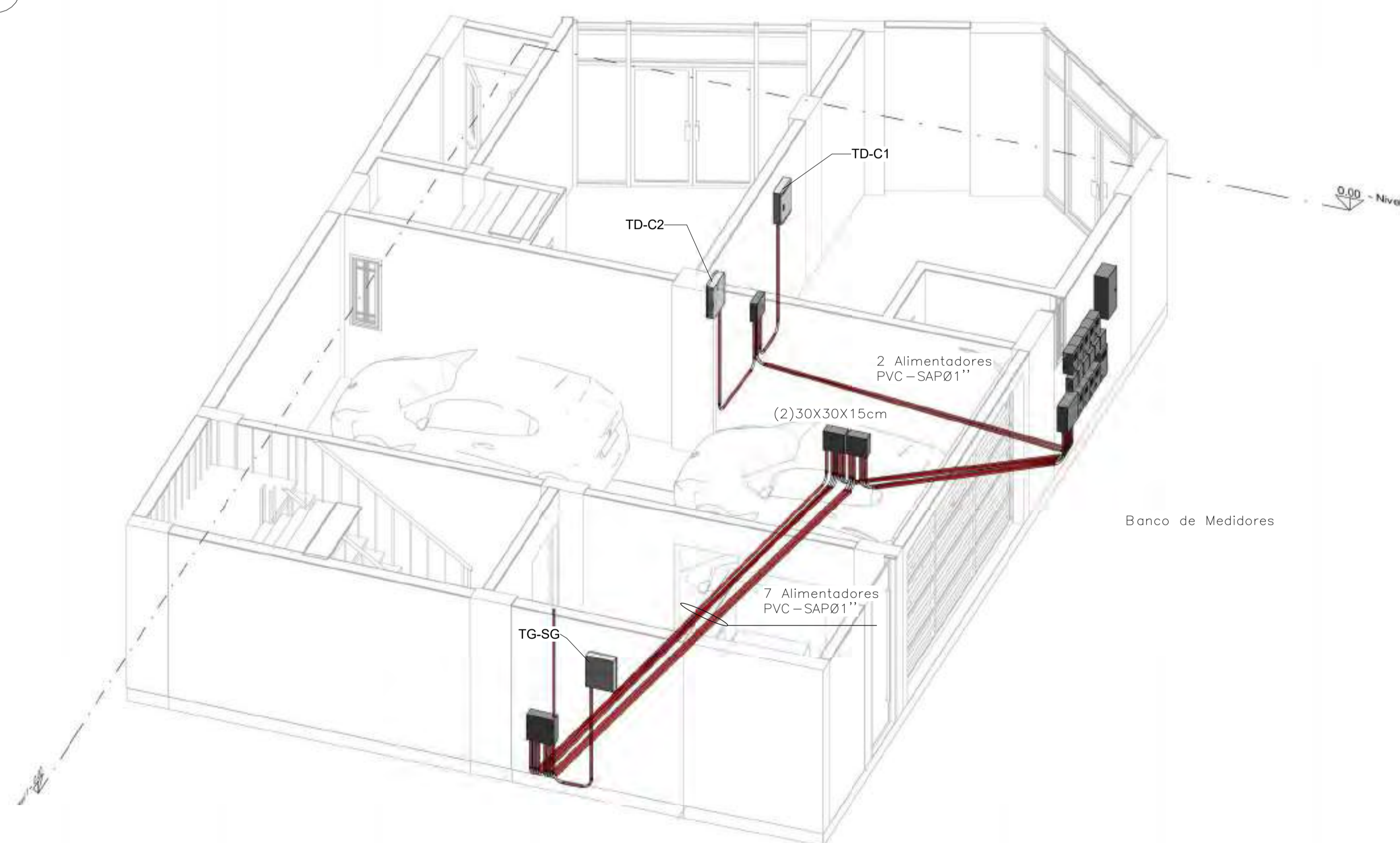
LEYENDA ELÉCTRICA

SÍMBOLO	DESCRIPCIÓN	ALTURA DE MONTAJE
	Medidor Eléctrico	Ver detalle
	Caja de paso metálica: a: 30X30X10cm b: 30X30X15cm	0.4m SNPT
	Caja de paso metálica: c: 40X40X15cm	0.4m SNPT
	Caja matriz de acometida eléctrica F1: 32X67X20cm	Ver detalle
	Circuito Alimentador eléctrico en conducto PVC-SAPØ25mm2 salvo indicación en plano	Suelo
	Tablero general o de distribución Monofásico o trifásico 220V 60Hz	1.60m SNPT

Alimentadores Eléctricos Primer Nivel  
1:50



Detalle Banco de Medidores  
1:25



Isometría - Alimentadores Eléctricos





UNIVERSIDAD  
NACIONAL DE  
SAN ANTONIO  
ABAD DEL  
CUSCO

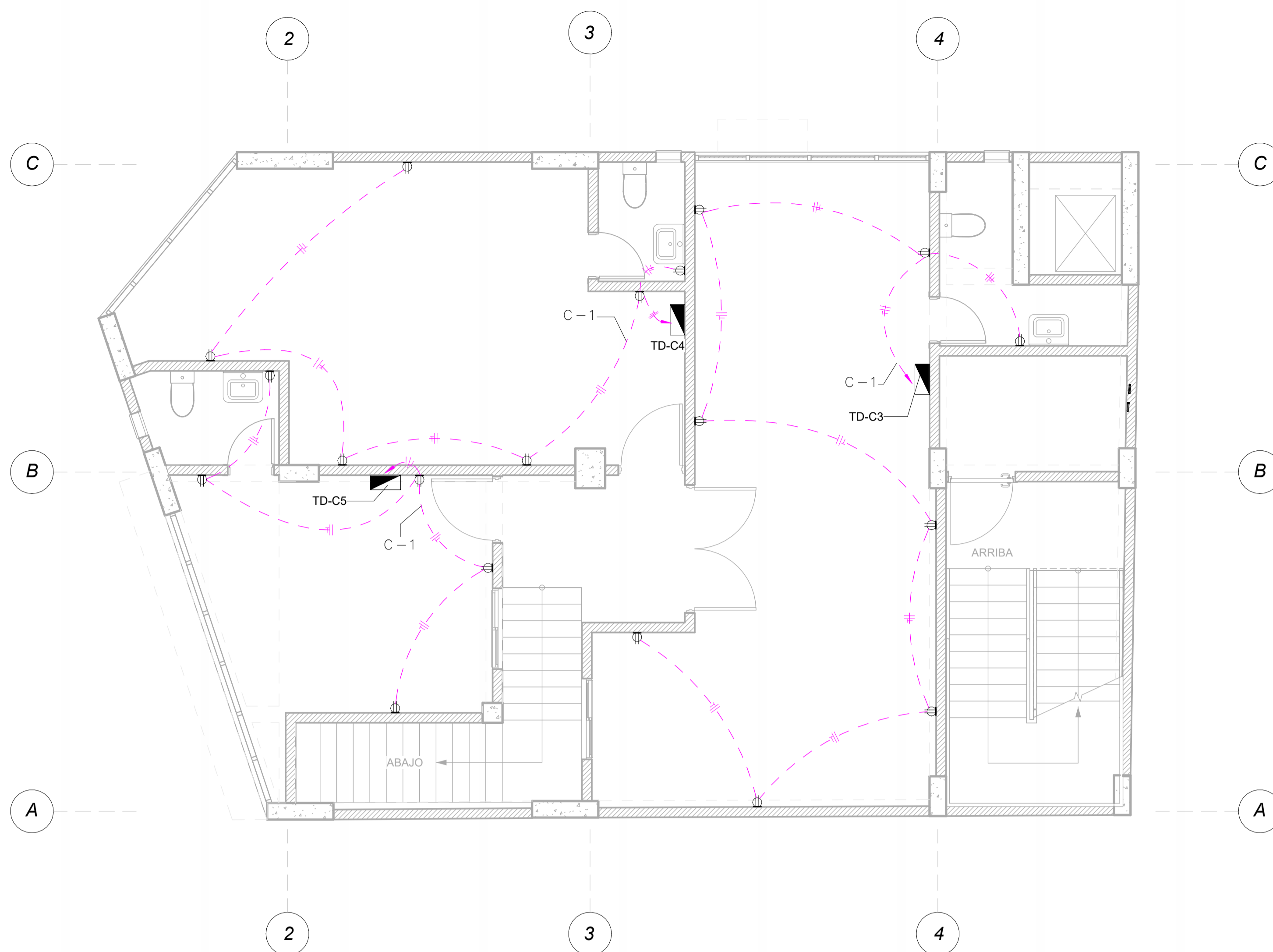
FACULTAD DE  
INGENIERÍA  
ELÉCTRICA,  
ELECTRÓNICA,  
INFORMÁTICA Y  
MECÁNICA

ESCUELA  
PROFESIONAL DE  
INGENIERÍA  
ELÉCTRICA

Tomacorrientes  
2do Nivel

LÁMINA:

IE-02



CUADRO DE CARGAS TD-C3

Norma	Carga	Descripción	Potencia	FD	Máxima Demanda
S050-210(a)	Carga Básica	33.64m2 (25W/m2)	841		841
		<b>Total</b>	<b>841</b>		<b>841</b>
	Carga Adicional		0		0
			0		0
	Apl. de Factores Acometida		841	1	841
	Alimentador		841	1	841
		<b>Total</b>	<b>841</b>		<b>841</b>

CUADRO DE CARGAS TD-C4

Norma	Carga	Descripción	Potencia	FD	Máxima Demanda
S050-210(a)	Carga Básica	31.32m2 (25W/m2)	783		783
		<b>Total</b>	<b>783</b>		<b>783</b>
	Carga Adicional		0		0
			0		0
	Apl. de Factores Acometida		783	1	783
	Alimentador		783	1	783
		<b>Total</b>	<b>783</b>		<b>783</b>

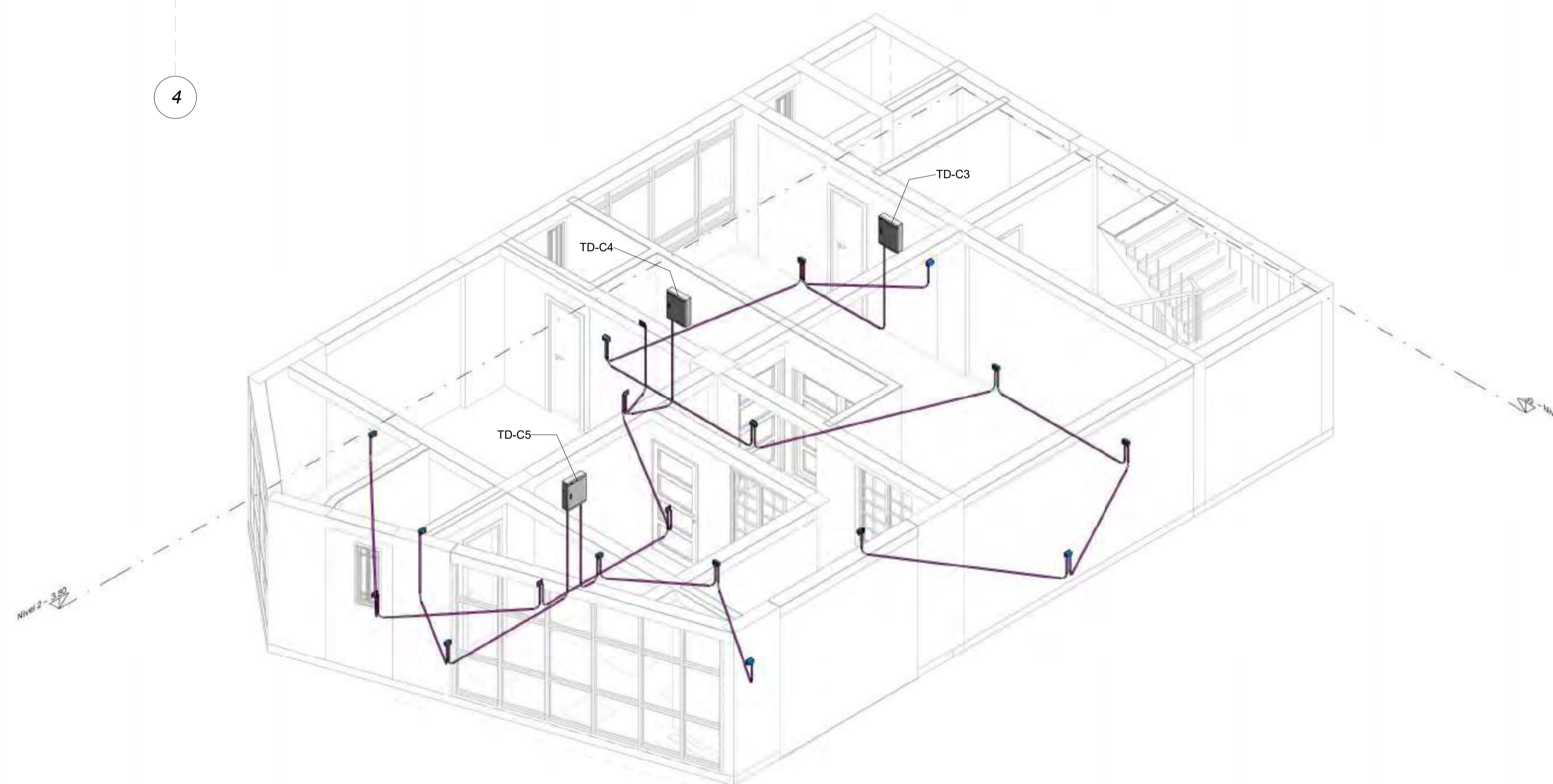
Circuito de tomacorrientes - Nivel 2  
1:50

CUADRO DE CARGAS TD-C5

Norma	Carga	Descripción	Potencia	FD	Máxima Demanda
S050-210(a)	Carga Básica	19.21m2 (25W/m2)	480.25		480.25
		<b>Total</b>	<b>480.25</b>		<b>480.25</b>
	Carga Adicional		0		0
			0		0
	Apl. de Factores Acometida		480.25	1	480.25
	Alimentador		480.25	1	480.25
		<b>Total</b>	<b>480.25</b>		<b>480.25</b>

LEYENDA ELÉCTRICA

SÍMBOLO	DESCRIPCIÓN	ALTURA DE MONTAJE
	Tablero general o de distribución Monofásico o trifásico 220V 60Hz	1.60m SNPT
	Salida para tomacorrientes universal 220V 60Hz	0.40m SNPT (Salvo indicación en plano)
	Circuito de tomacorrientes en conducto PVC-SAP820mm2 salvo indicación en plano	Suelo



Isometría - Tomacorrientes 2do Nivel



UNIVERSIDAD  
NACIONAL DE  
SAN ANTONIO  
ABAD DEL  
CUSCO

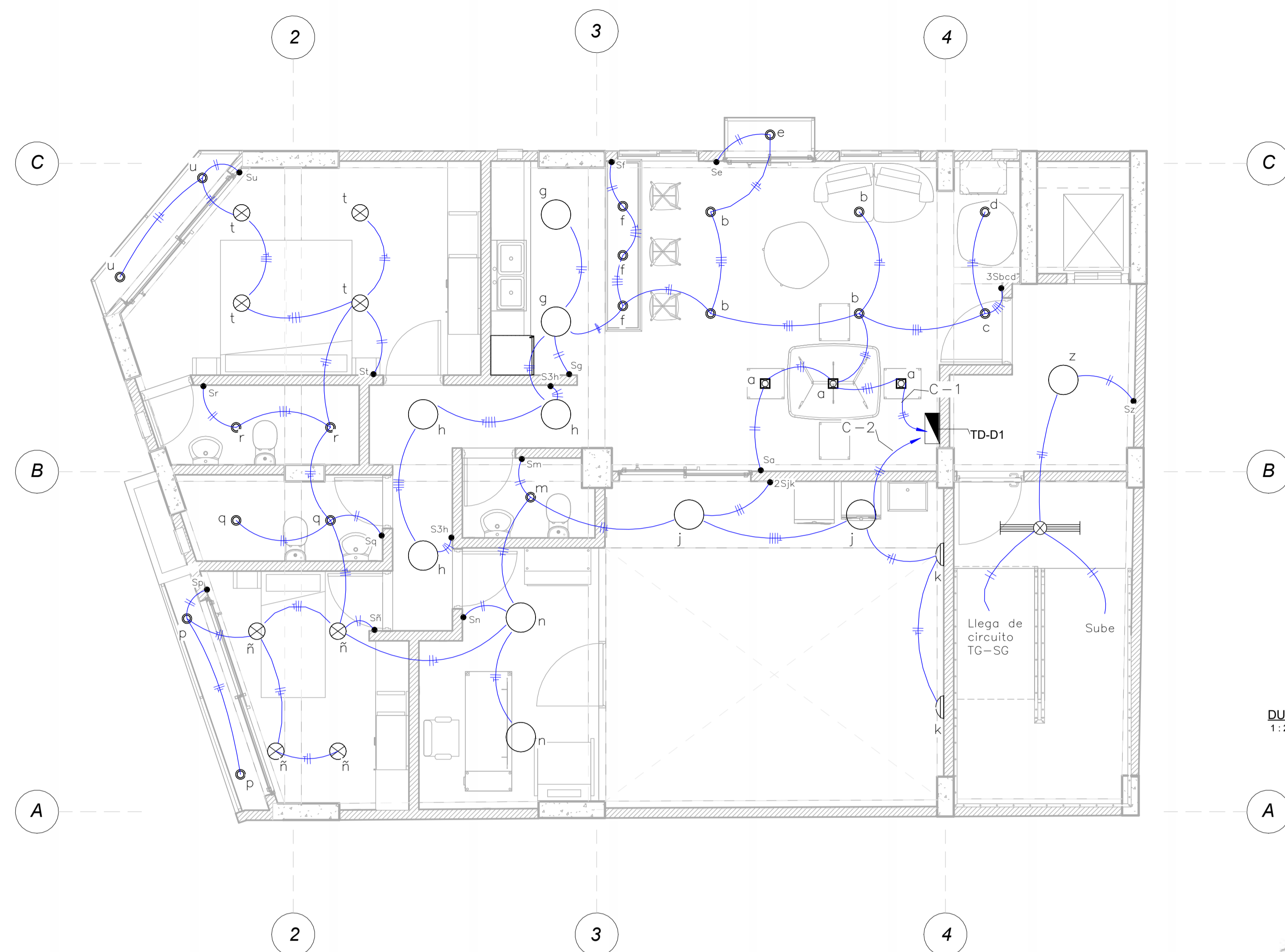
FACULTAD DE  
INGENIERÍA  
ELÉCTRICA,  
ELECTRÓNICA,  
INFORMÁTICA Y  
MECÁNICA

ESCUELA  
PROFESIONAL DE  
INGENIERÍA  
ELÉCTRICA

Alumbrado  
3er Nivel

LÁMINA:

IE-03

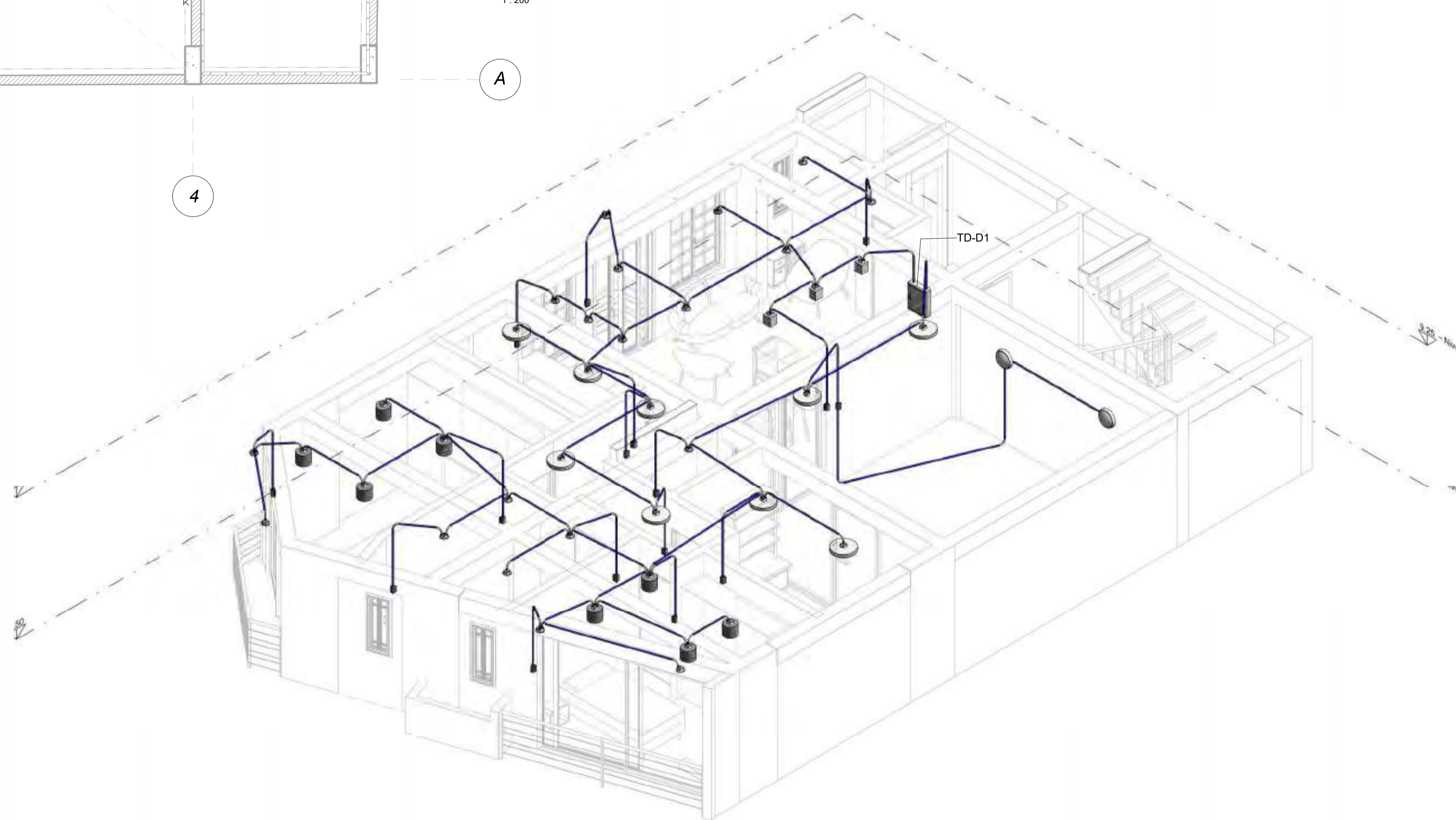


DU\_TD-01  
1:200

IE\_Ilu\_Nivel 3  
1:50

LEYENDA ELÉCTRICA

SÍMBOLO	DESCRIPCIÓN	ALTURA DE MONTAJE
	Tablero general o de distribución Monofásico o trifásico 220V 60Hz	1.60m SNPT
	LEDVANCE Downlight confort 13W 4000K 220V 60Hz 1210Lm o similar	Techo
	TRILUX Onplana 32W 6500K 220V 60Hz 1000Lm o similar	Techo
	TRILUX Dicroico Cuadrado SQ 10W 4000K 220V 60Hz 800Lm o similar	Techo
	PHILIPS Flexblend LED 48W 6500K 220V 60Hz 1800Lm o similar	Techo
	PHILIPS Luxspace 16W 4000K 220V 60Hz 800Lm o similar	Techo
	Interruptor eléctrico tipo: - Simple - Doble - Conmutado Simple	1.40m SNPT
	Circuito de Alumbrado en conducto PVC-SAPØ20mm2 salvo indicación en plano	Suelo



Isometría - Alumbrado 3er Nivel



UNIVERSIDAD  
NACIONAL DE  
SAN ANTONIO  
ABAD DEL  
CUSCO

FACULTAD DE  
INGENIERÍA  
ELÉCTRICA,  
ELECTRÓNICA,  
INFORMÁTICA Y  
MECÁNICA

ESCUELA  
PROFESIONAL DE  
INGENIERÍA  
ELÉCTRICA

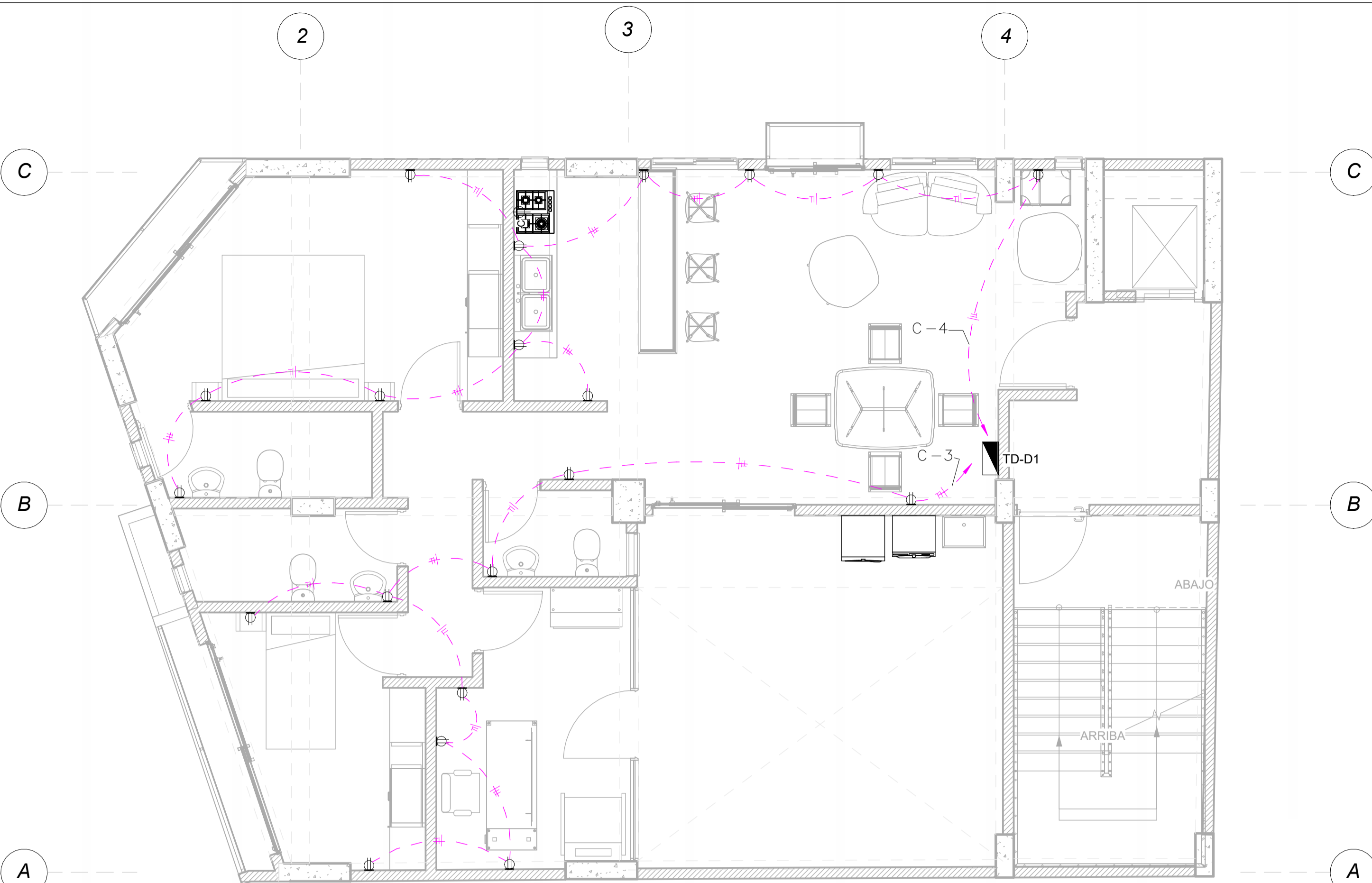
Tomacorrientes  
3er Nivel y  
caídas de  
tensión

LÁMINA:

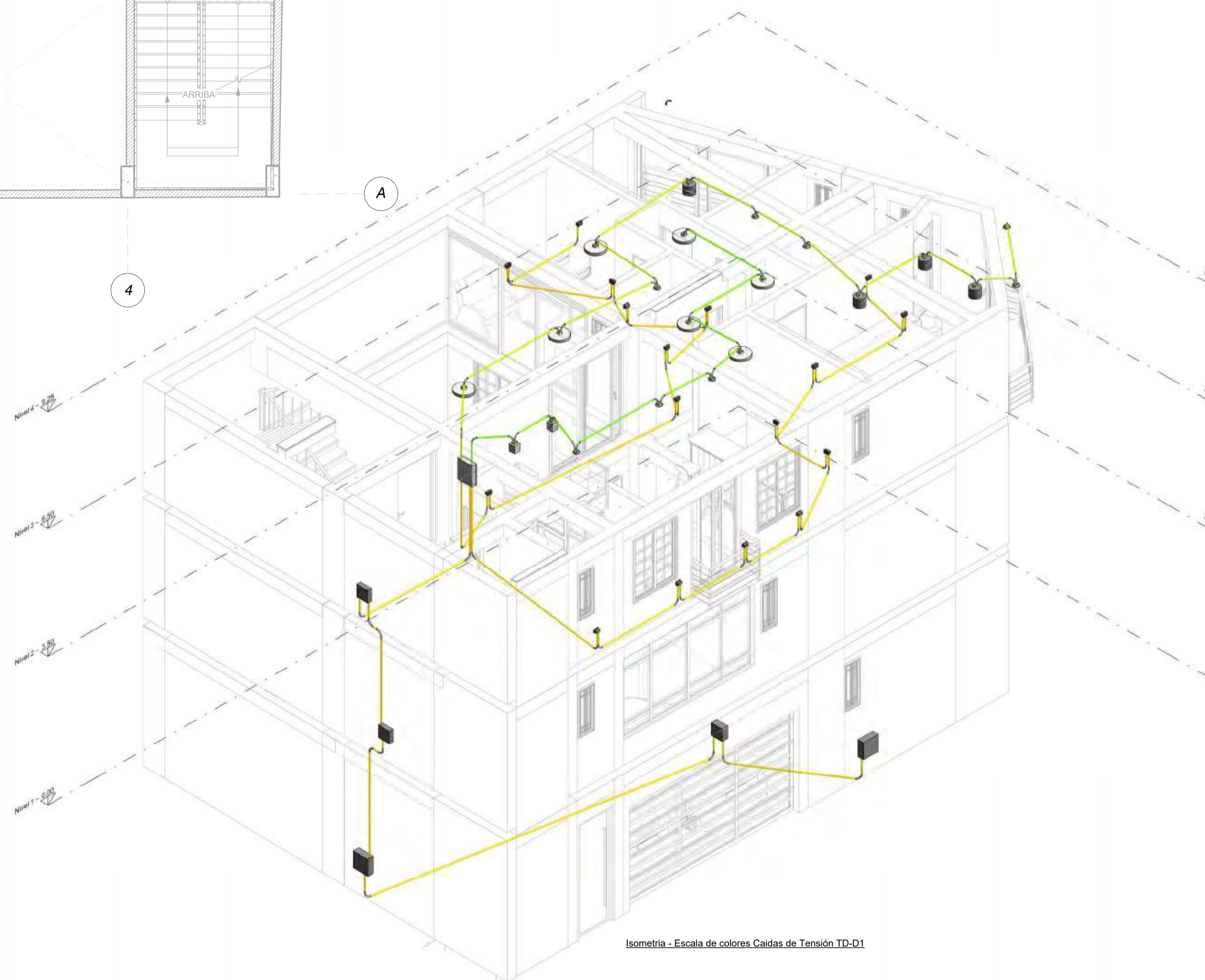
IE-04

CUADRO DE  
CARGAS TD-D1

Norma	Carga	Descripción	Potencia	FD	Máxima Demanda
S050-202(1) (a)(i)(ii)(iii)	Carga Básica	81.11m2 techados			
		Primeros 45m2	1500	1	1500
		Segundos 45m2	1000	1	1000
		Restante por cada 90m2	0	1	0
		<b>Total</b>	<b>2500</b>		<b>2500</b>
S050-202(1) (a)(iv)	Carga de Cocina Eléctrica	Potencia Base	0		0
		Adicional a 12KW	0	0.4	0
		<b>Total</b>	<b>0</b>		<b>0</b>
S050-202(1) (a)(vi)	Carga Adicional		0	1	0
			0	1	0
		<b>Total</b>	<b>0</b>		<b>0</b>
S050-202(1) (a)(v)	Aire Acondicionado		0	1	0
			0		0
		<b>Total</b>	<b>0</b>		<b>0</b>
		<b>Total</b>	<b>2500</b>		<b>2500</b>



Circuito de tomacorrientes - Nivel 3  
1:50



Isometría - Escala de colores Caídas de Tensión TD-D1

LEYENDA ELÉCTRICA

SÍMBOLO	DESCRIPCIÓN	ALTURA DE MONTAJE
	Tablero general o de distribución Monofásico o trifásico 220V 60Hz	1.60m SNPT
	Salida para tomacorrientes universal 220V 60Hz	0.40m SNPT (Salvo indicación en plano)
	Circuito de tomacorrientes en conducto PVC-SAPØ20mm2 salvo indicación en plano	Suelo

Leyenda IE\_Tomacorrientes  
1:50

Circuito	Corriente de Diseño	Longitud	Material	Sección	DV	DV%
AD1-1	26.85	22.15	Cobre	10	1.884	0.856
C-1	1.82	16.59	Cobre	2.5	0.382	0.174
C-2	2.69	32.73	Cobre	2.5	1.115	0.507
C-3	10.23	23.40	Cobre	4	1.895	0.862
C-4	12.50	20.11	Cobre	4	1.991	0.905

**Anexo E: Manual de usuario de la propuesta**

El presente anexo expone el manual de usuario elaborado para la correcta ejecución de la propuesta y detalles técnicos concernientes a su instalación y reporte de errores.

# Scripts de asistencia al diseño y cálculo de instalaciones electricas interiores BIM Manual de usuario v. 2023-1

Por: Luis Giuseppe Yucra Altamirano

2022-2023



## Resumen

El siguiente manual contiene una guía completa de uso para la propuesta elaborada en la tesis titulada “*Programación paramétrica–visual aplicada a software BIM para el desarrollo de la etapa de diseño y cálculo de instalaciones eléctricas interiores.*”, consultar el repositorio de la Universidad Nacional de San Antonio Abad del Cusco para mayor información.

# Índice

<b>1. Introducción e instalación</b>	<b>3</b>
1.1. Estructura del Manual . . . . .	3
1.2. Consideraciones de compatibilidad . . . . .	3
1.3. Requerimientos . . . . .	3
1.4. Instalación . . . . .	3
1.5. En el paquete digital . . . . .	4
1.6. Antes de comenzar . . . . .	4
<b>2. Sobre los scripts</b>	<b>5</b>
2.1. Compatibilidad con la Versión en Español de Revit . . . . .	5
2.2. Norma Peruana y Parámetros Específicos . . . . .	5
2.3. Instalación de Paquetería Adicional . . . . .	5
2.4. Política de código abierto . . . . .	5
2.4.1. Sobre los créditos . . . . .	6
2.4.2. Uso Responsable . . . . .	6
<b>3. Primeros pasos en Revit</b>	<b>7</b>
3.1. Parámetros Nuevos . . . . .	7
3.2. Manejo de unidades . . . . .	8
3.2.1. Coherencia de Unidades en Revit . . . . .	8
3.3. Primera ejecución . . . . .	8
<b>4. Uso de los scripts</b>	<b>10</b>
4.1. Orden correcto para la ejecución de los scripts . . . . .	10
4.2. Guía de uso por script . . . . .	10
4.2.1. Script Conductos i.dyn . . . . .	10
4.2.2. Script Iluminacion optima.dyn . . . . .	11
4.2.3. Script conductos I.dyn . . . . .	12
4.2.4. Script Tomacorrientes.dyn . . . . .	13
4.2.5. Script Conductos II.dyn . . . . .	13
4.2.6. Script Cuadros de carga.dyn . . . . .	14
4.2.7. Script Caidas de tension.dyn . . . . .	14
<b>5. Scripts Custom</b>	<b>16</b>
5.1. ¿Cómo editar los scripts? . . . . .	16
<b>6. Preguntas Frecuentes</b>	<b>17</b>
<b>7. Acerca del desarrollador</b>	<b>18</b>

# 1. Introducción e instalación

Bienvenido al Manual de Usuario de los Scripts para Ingeniería Eléctrica en Plataforma BIM.

Este manual está diseñado para proporcionarte una comprensión completa y práctica de las herramientas y scripts incluidos en esta herramienta, los cuales están destinados a optimizar tus proyectos de infraestructura eléctrica dentro del entorno de Building Information Modeling (BIM).

Para aprovechar al máximo la funcionalidad de las herramientas descritas en este manual, te recomendamos profundizar en tus conocimientos sobre BIM. Esto te permitirá utilizar estas herramientas de manera efectiva y potenciar tus proyectos de infraestructura eléctrica en el contexto BIM.

## 1.1. Estructura del Manual

Este manual se divide en secciones que te guiarán a través de la instalación, configuración y uso de los scripts. Cada sección proporciona instrucciones detalladas y ejemplos para ayudarte a dominar estas herramientas.

## 1.2. Consideraciones de compatibilidad

Los scripts fueron desarrollados en la base programable Dynamo versión 2, por lo que su compatibilidad es únicamente con Revit 2022 o inferiores.

Sobre versiones superiores de Revit se recomienda un manejo cauteloso de la herramienta, ya que esta presta funcionalidad de nodos desarrollados por terceros.

No es posible ejecutar la propuesta en otros softwares BIM aparte de Revit (por ejemplo: ArchiCAD, All Plan, Bentley, etc)

## 1.3. Requerimientos

Se recomienda basarse en los requerimientos mínimos de Autodesk Revit 2022 para poder lograr el mejor rendimiento.

## 1.4. Instalación

Antes de comenzar a utilizar los archivos y scripts proporcionados en este paquete digital, asegúrate de que tu sistema cumple con los siguientes requisitos:

- **Revit 2022:** Este paquete ha sido diseñado para ser compatible con Autodesk Revit 2022. Asegúrate de tener Revit 2022 instalado en tu sistema antes de proceder. Se ha comprobado el funcionamiento tanto en licencias definitivas como de estudiante.
- **Dynamo:** Dynamo es una herramienta que funciona como un complemento para Revit y permite ejecutar scripts de diseño paramétrico. Al tener Revit 2022 instalado, Dynamo debería estar incluido en tu instalación de Revit. Asegúrate de tenerlo habilitado y listo para su uso.
- **Python:** Python es un lenguaje de programación utilizado en la creación y modificación de scripts dentro de Dynamo. Es importante tener Python instalado en tu sistema para

acceder y editar el código fuente de los scripts, si así lo deseas. Puedes descargar Python desde el sitio web oficial de Python (<https://www.python.org>).

- **Dialux Evo y AutoCAD:** Se recomienda la instalación de Dialux Evo para análisis de iluminación y AutoCAD u otro software con capacidad para leer archivos DWG para la colaboración y lectura de archivos de dibujo técnico.

## 1.5. En el paquete digital

El paquete digital contiene los archivos necesarios para ejecutar los scripts. Estos archivos se componen principalmente de archivos con extensión `.dyn` y un archivo `.lsp`.

- `0_Conductos i.dyn`
- `1_IluminacionOptima.dyn`
- `2_Conductos I.dyn`
- `3_Tomacorrientes.dyn`
- `4_Conductos II.dyn`
- `5_Cuadros de carga.dyn`
- `6_Caidas de tension.dyn`
- `Export block coordinates.lsp`

A diferencia de algunos complementos o extensiones que requieren instalación en carpetas específicas, estos archivos no necesitan ser instalados en ningún lugar específico en tu sistema.

Puedes guardar los archivos `.dyn` y `.lsp` en cualquier ubicación de tu elección, ya sea en tu disco duro local, una unidad externa o en la nube. Lo importante es que puedas acceder fácilmente a ellos cuando los necesites para ejecutar los scripts.

## 1.6. Antes de comenzar

Además de los requisitos de software, es altamente recomendable tener un conocimiento básico en los siguientes aspectos:

- **Lenguaje Dynamo:** Para utilizar eficazmente los scripts y personalizarlos según tus necesidades, es beneficioso tener un conocimiento básico en el lenguaje Dynamo. Esto te permitirá comprender la lógica detrás de los scripts y realizar ajustes según sea necesario.
- **Python:** Si planeas modificar el código fuente de los scripts, es esencial tener un conocimiento básico de Python. Esto te permitirá comprender y editar el código de manera efectiva.

Con estos requisitos y conocimientos recomendados, estarás listo para aprovechar al máximo las herramientas y scripts contenidos en este paquete digital en tu entorno Revit 2022.



## 2. Sobre los scripts

Los scripts incluidos en este paquete digital han sido diseñados para funcionar de manera óptima en la versión en español de Autodesk Revit, siguiendo las normas y estándares peruanos aplicables. Además, han sido creados con la adición de parámetros específicos en Revit para satisfacer las necesidades particulares de proyectos bajo estas condiciones.

### 2.1. Compatibilidad con la Versión en Español de Revit

Es importante destacar que los scripts han sido desarrollados y probados en la versión en español de Autodesk Revit. Si estás utilizando una versión diferente de Revit o un idioma diferente, es posible que algunas funciones no se comporten de la misma manera. Se recomienda encarecidamente utilizar la versión en español de Revit para garantizar la compatibilidad y el funcionamiento óptimo de los scripts.

### 2.2. Norma Peruana y Parámetros Específicos

Los scripts han sido configurados teniendo en cuenta las normas y regulaciones peruanas relevantes para proyectos de infraestructura. Asimismo, se han incorporado parámetros específicos en Revit que son comunes en proyectos de construcción y diseño en Perú.

Al utilizar estos scripts, puedes aprovechar la automatización y las capacidades de diseño paramétrico dentro del contexto de las normas peruanas y los parámetros personalizados, lo que facilita la adaptación de los proyectos a los requisitos locales.

Para obtener detalles sobre la configuración de parámetros adicionales en Revit, te recomendamos consultar la sección correspondiente en el propio software. Esta sección proporcionará información detallada sobre cómo configurar y personalizar los parámetros según las necesidades específicas de tu proyecto. Sin embargo, en este manual también se te proveerá los parámetros con los que los scripts funcionan para tareas de generación de cálculo.

### 2.3. Instalación de Paquetería Adicional

Para aprovechar al máximo la funcionalidad de estos scripts, es posible que necesites instalar paquetes adicionales en Dynamo. No te preocupes, no es necesario realizar una instalación profunda. Al abrir Dynamo, el software reconocerá automáticamente los paquetes faltantes y te permitirá instalarlos según sea necesario.

### 2.4. Política de código abierto

Los scripts incluidos en este paquete digital son herramientas de código abierto diseñadas para brindar flexibilidad y eficiencia en el desarrollo de proyectos de infraestructura eléctrica en el entorno de Building Information Modeling (BIM). Como parte de nuestra filosofía de colaboración y conocimiento compartido, estos scripts se ponen a disposición de la comunidad de usuarios para su libre manipulación, uso y mejora.

Entendemos la importancia de la adaptación y personalización de herramientas para satisfacer las necesidades específicas de cada proyecto. Por lo tanto, alentamos a los usuarios a modificar y ajustar los scripts según sea necesario para lograr los objetivos deseados. No hay restricciones en la manipulación y modificación de los scripts dentro de los límites de la licencia de código abierto bajo la cual se distribuyen.

#### **2.4.1. Sobre los créditos**

Aunque se permite la modificación y el uso de los scripts de forma libre, se solicita encarecidamente a los usuarios que, en caso de realizar modificaciones sustanciales o utilizar estos scripts en proyectos públicos o compartidos, proporcionen los créditos correspondientes al autor original. Este reconocimiento es importante para honrar el esfuerzo y la contribución de quien ha trabajado en la creación y mejora de estas herramientas.

Se recomienda consultar el repositorio oficial de la Universidad Nacional de San Antonio Abad del Cusco para obtener más información sobre el autor.

#### **2.4.2. Uso Responsable**

Si bien se fomenta la libertad de manipulación de los scripts, también se espera que los usuarios utilicen estas herramientas de manera responsable y ética. La adaptación y el uso de los scripts deben estar en conformidad con las leyes y regulaciones aplicables.

A través de esta política, buscamos fomentar un entorno colaborativo y de respeto mutuo entre los usuarios de estas herramientas de código abierto. Al compartir conocimientos y reconocer las contribuciones, podemos continuar mejorando y optimizando nuestras prácticas en el campo de la ingeniería y el diseño en el entorno BIM.

## 3. Primeros pasos en Revit

En esta sección, se proporcionan detalles importantes sobre la implementación de nuevos parámetros en Autodesk Revit, específicamente relacionados con la disciplina eléctrica. Estos parámetros son esenciales para una documentación precisa de los cálculos asistidos por Dynamo.

### 3.1. Parámetros Nuevos

A continuación, se detallan los parámetros que son necesarios implementar en Revit, especialmente en la especialidad eléctrica. Estos parámetros permiten una descripción detallada de los elementos eléctricos y son fundamentales para un diseño y documentación precisos:

- **Norma:**  
Este parámetro hace referencia a la norma aplicable para el cuadro de cargas relacionado con los elementos eléctricos. Es importante incluir esta información para cumplir con las regulaciones y normativas específicas.
- **Descripción:**  
El parámetro “Descripción” se utiliza para proporcionar una descripción detallada de la carga eléctrica asociada con los elementos. Esta descripción ayuda a comprender la naturaleza y el propósito de la carga.
- **FD (Factor de Demanda):**  
El factor de demanda es un valor que representa la carga eléctrica esperada en relación con la carga máxima en un momento dado. Su inclusión es crucial para el dimensionamiento adecuado de los sistemas eléctricos y la gestión de la demanda.
- **PI (Potencia Instalada):**  
Este parámetro indica la potencia total instalada de los elementos eléctricos. Proporciona información sobre la capacidad de carga y es esencial para la planificación y el cálculo de la distribución de energía.
- **MD (Máxima Demanda):**  
La máxima demanda representa la carga eléctrica máxima registrada durante un período específico. Conocer este valor es fundamental para diseñar sistemas eléctricos que puedan manejar picos de demanda.

Además de los parámetros previamente mencionados, es crucial considerar la implementación de dos parámetros adicionales relacionados con el cálculo de caídas de tensión en tu proyecto de Revit. Estos parámetros son esenciales para evaluar y gestionar la calidad del suministro eléctrico en tus diseños.

- **Conductor Corregido ( $mm^2$ ):**  
Aunque el script no calcula automáticamente el conductor utilizado en Revit, se recomienda encarecidamente la inclusión de este parámetro. El “Conductor Corregido” se utiliza para especificar el tamaño del conductor eléctrico en milímetros cuadrados ( $mm^2$ ). Al registrar esta información, podrás evaluar la capacidad del conductor y su idoneidad para transportar la corriente eléctrica requerida sin pérdida significativa de tensión. Esto es fundamental para garantizar un suministro eléctrico eficiente y evitar problemas de caídas de tensión.

- **DV** (Caída de Tensión):

El parámetro “DV” expresa la caída de tensión en el sistema eléctrico y es un indicador crítico de la calidad del suministro eléctrico. Representa la diferencia de tensión entre el punto de suministro y el punto de consumo. Su inclusión permite evaluar si la tensión es suficiente para los dispositivos eléctricos conectados y si se cumplen los estándares de calidad eléctrica. El valor de "DV" debe calcularse de acuerdo con las especificaciones del proyecto y las normas aplicables.

**IMPORTANTE:** Todos estos parámetros deben ser creados como **Texto** dentro de Revit 2022, esto permite que los scripts dedicados al cálculo puedan funcionar correctamente.

## 3.2. Manejo de unidades

El uso de unidades de medida coherentes es esencial para garantizar la precisión y la consistencia en tus proyectos de infraestructura eléctrica en Revit. Se recomienda encarecidamente el uso de unidades de medida estándar recomendadas por organismos como el Sistema Legal de Unidades de Medida Peruanas (SLUMP). Estas unidades estandarizadas facilitan la comunicación y la interpretación de los datos en el ámbito de la ingeniería eléctrica.

### 3.2.1. Coherencia de Unidades en Revit

Al trabajar en proyectos de infraestructura eléctrica en Revit, es crucial mantener la coherencia de unidades en todo el proyecto. Esto significa que todas las propiedades y parámetros deben tener las mismas unidades de medida. La falta de coherencia en las unidades puede conducir a errores de cálculo y complicaciones en la ejecución de los scripts.

## 3.3. Primera ejecución

Antes de comenzar a utilizar los scripts y herramientas proporcionados, es fundamental tener en cuenta un proceso inicial que garantizará un funcionamiento sin problemas y la actualización de los paquetes necesarios. Sigue estas recomendaciones en tu primera ejecución.

### **OJO: Abre los Scripts a través de Dynamo, no del Reproductor Dynamo**

En tu primera ejecución, te recomendamos encarecidamente abrir los scripts directamente a través de la plataforma Dynamo en lugar de utilizar el Reproductor Dynamo. Esta práctica es fundamental para asegurarte de que se actualicen automáticamente los paquetes faltantes que pueden ser necesarios para el funcionamiento correcto de los scripts.

Al abrir los scripts desde Dynamo, el software detectará automáticamente los paquetes que faltan y te permitirá instalarlos de manera eficiente. Esto garantiza que todas las dependencias requeridas estén disponibles y que los scripts funcionen sin problemas.

Sigue estos pasos para abrir los scripts desde Dynamo:

1. Abre Autodesk Revit 2022 en tu sistema.
2. Accede a la pestaña “Gestionar” o “Add-Ins” en Revit.
3. Selecciona "Dynamo" para abrir la plataforma.

4. Carga el archivo `.dyn` correspondiente al script que deseas ejecutar utilizando la opción de carga de archivo dentro de Dynamo.
5. Verificar los mensajes de paquetería faltante.
6. Instalar los paquetes que sean necesarios. No temer por virus, esta instalación se hace a través del repositorio oficial de Dynamo, en donde se aseguran que todos los nodos y funciones no contengan malware.

Siguiendo este enfoque, estarás listo para utilizar los scripts de manera efectiva en tu proyecto de infraestructura eléctrica en plataforma BIM.

## 4. Uso de los scripts

### 4.1. Orden correcto para la ejecución de los scripts

En el entorno de los scripts para el diseño y cálculo de instalaciones eléctricas en Revit 2022, es importante destacar que no existe un orden predefinido para la ejecución de scripts. Estas herramientas están diseñadas para adaptarse a las necesidades específicas de cada usuario y pueden ser utilizadas en cualquier etapa del desarrollo de su proyecto. Sin embargo, con el propósito de facilitar la experiencia del usuario y garantizar una ejecución más eficiente de los scripts, se sugiere seguir el diagrama de flujo mostrado en la siguiente figura.

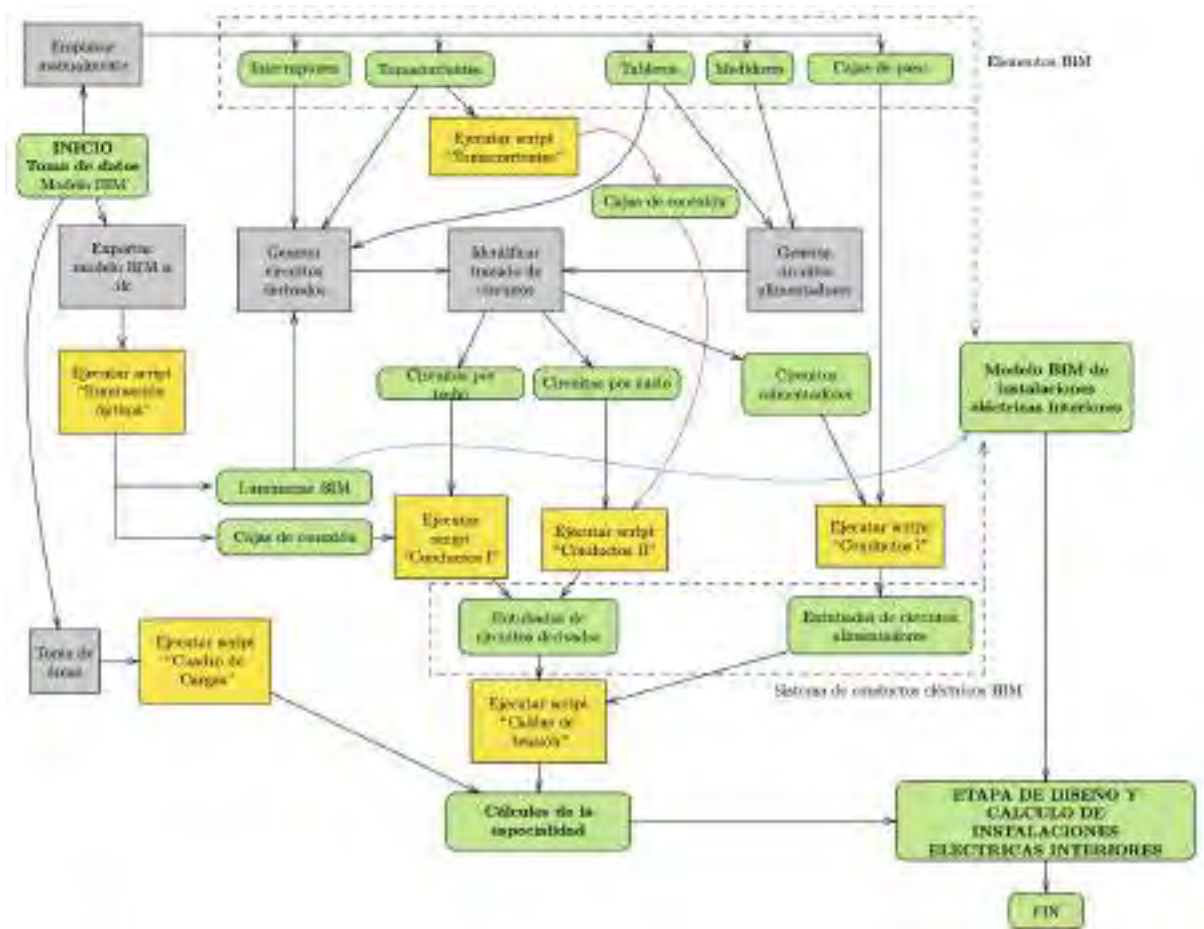


Figura 1: Diagrama de flujo sugerido para la ejecución de los scripts

Este diagrama de flujo proporciona una guía visual que muestra una secuencia lógica recomendada para la ejecución de los scripts. Si bien no es un requisito estricto, seguir esta secuencia puede ayudarte a maximizar la eficacia de los scripts y garantizar que tus proyectos se desarrollen de manera más ordenada y organizada.

### 4.2. Guía de uso por script

#### 4.2.1. Script Conductos i.dyn

Este script está dedicado a facilitar la creación de un entubado por suelo con la capacidad de cambiar de nivel, lo que representa las montantes eléctricas en niveles BIM. Para

ejecutarlo correctamente, sigue los siguientes pasos:

1. **Ejecutar el script:** Abre el reproductor de Dynamo y carga el script [Conductos i.dyn](#).
2. **Interfaz de selección:** Una vez que el script esté en ejecución, deberás seleccionar en orden secuencial la ruta del entubado. Es importante estudiar cuidadosamente las rutas para evitar ocupar conectores cercanos a cajas de paso, tableros o subtableros.
3. **Seleccionar un tubo de referencia:** Deberás seleccionar un tubo de referencia que servirá como base para clonar y definir las propiedades del nuevo entubado.
4. **Indicar el diámetro:** Especifica el diámetro del tubo. Asegúrate de que esté en conformidad con las unidades configuradas en Revit para garantizar la precisión del diseño.
5. **Presionar el botón “Entubar”:** Una vez que hayas seleccionado la ruta, el tubo de referencia y configurado el diámetro, presiona el botón “Entubar” para que el script genere el entubado de acuerdo a tus especificaciones.
6. **Confirmación de la conexión (opcional):** En algunas versiones de Revit, es necesario confirmar la conexión de cada ruta de los tubos. Puedes verificar el enrutado presionando la tecla “Tab” manteniendo el cursor sobre algún elemento del enrutado. Esto garantizará que el entubado esté correctamente conectado y funcionando como se espera.

Es importante recordar que este script ofrece la flexibilidad de adaptarse a tus necesidades específicas y las particularidades de tu proyecto. No dudes en ajustar los parámetros y personalizar la creación del entubado según lo que requiera tu diseño eléctrico en Revit 2022.

#### 4.2.2. Script Iluminacion optima.dyn

Este script está diseñado para facilitar el emplazamiento automático de luminarias en base a un cálculo lumínico previo. Para utilizarlo de manera efectiva, sigue estos pasos:

##### 1. En Revit:

- a) Exporta tu modelo BIM en un formato IFC básico.

##### 2. En Dialux:

- a) Importa el modelo IFC en el software Dialux.
- b) Configura los espacios o locales con los requisitos mínimos y comienza a emplazar las luminarias y ejecutar el cálculo necesario para satisfacer las necesidades de iluminación.
- c) Una vez satisfecho con el diseño de iluminación, exporta los resultados en formato DWG. Se recomienda realizar la exportación de la siguiente manera:
  - Crea una capa separada para cada tipo de luminaria.
  - Exporta únicamente los contornos de la edificación.
  - No exportes las líneas isolux.

##### 3. En AutoCAD:

- a) Debes importar el archivo `export block coordinates.lsp` en AutoCAD utilizando el comando `APPLOAD`.
- b) Abre el archivo generado por Dialux en AutoCAD.
- c) Ejecuta el comando `EXPORTBLOCKCOORDINATES` y selecciona todas las luminarias en el dibujo.
- d) En la carpeta del archivo CAD, encontrarás un archivo de texto básico con información sobre las ubicaciones de las luminarias. Puedes cambiar el nombre de este archivo si lo deseas, ya que no afectará el funcionamiento del script.

#### 4. En Revit:

- a) Ejecuta el script `Illuminacion optima.dyn`.
- b) Debes ubicar y seleccionar el archivo de texto generado anteriormente en AutoCAD.
- c) Selecciona el tipo de luminaria que deseas utilizar. Asegúrate de que la luminaria elegida sea la misma o similar a las utilizadas en Dialux.
- d) Si es necesario, puedes emplazar las cajas octogonales para cada luminaria.
- e) Elige una superficie de referencia para ubicar las luminarias.
- f) Presiona el botón `.Emplazar.en` la interfaz del script.

Este proceso te permitirá importar y emplazar automáticamente las luminarias en tu proyecto de Revit, basado en el diseño lumínico calculado en Dialux. Asegúrate de seguir cada paso cuidadosamente para obtener los resultados deseados.

#### 4.2.3. Script conductos I.dyn

Este script está diseñado para facilitar la creación de entubados por techo, y se recomienda su uso en circuitos de alumbrado. Su funcionamiento es similar al script `Conductos i.dyn`. Sigue estos pasos para usarlo:

Ejecución del script:

1. Abre el reproductor de Dynamo y carga el script `Conductos I.dyn`.
2. **Interfaz de selección:** Una vez que el script esté en ejecución, sigue los mismos pasos que se indican en el script `Conductos i.dyn`. Deberás seleccionar en orden secuencial la ruta del entubado, teniendo en cuenta los mismos alcances de diseño, especialmente en lo que respecta al diámetro del tubo a utilizar.
3. **Utiliza la herramienta de circuitos automáticos de Revit:** Para generar las rutas más eficientes, se recomienda apoyarse en la herramienta de circuitos automáticos de Revit. Esto te permitirá generar las mejores rutas para tus entubados por techo.

Al seguir estos pasos, podrás crear entubados por techo para circuitos de alumbrado de manera eficiente. Como en el caso anterior, recuerda que puedes personalizar el script según las necesidades específicas de tu proyecto.



#### 4.2.4. Script Tomacorrientes.dyn

Este script es un proceso de diseño que facilita el emplazamiento de cajas rectangulares o de conexión necesarias para los tomacorrientes en tu proyecto. Sigue estos pasos para usarlo:

1. Asegúrate de haber colocado todos los tomacorrientes necesarios en tu proyecto antes de ejecutar el script.
2. Ejecución del script: Abre el reproductor de Dynamo y carga el script [Tomacorrientes.dyn](#).
3. **Selecciona todos los tomacorrientes:** Una vez que el script esté en ejecución, selecciona todos los tomacorrientes en los que consideres necesario incluir una caja de conexión.
4. Selecciona una o varias caras murales de referencia: Luego de seleccionar los tomacorrientes, elige una o varias caras murales de referencia donde deseas que se emplace la caja de conexión.
5. **Pulsa “Emplazar”:** Una vez que hayas seleccionado los tomacorrientes y las caras murales de referencia, presiona el botón “Emplazar” en la interfaz del script.

Siguiendo estos pasos, podrás diseñar de manera eficiente el emplazamiento de cajas de conexión para los tomacorrientes en tu proyecto. Recuerda que este script está diseñado para simplificar y agilizar el proceso de diseño, permitiéndote centrarte en otros aspectos importantes de tu proyecto eléctrico.

#### 4.2.5. Script Conductos II.dyn

Este script está diseñado para emplazar enrutados de circuitos por el suelo, sin considerar montantes. La ejecución de este script es similar a los scripts [Conductos i.dyn](#) y [Conductos I.dyn](#). Sigue estos pasos para utilizarlo:

1. **Ejecución del script:** Abre el reproductor de Dynamo y carga el script [Conductos II.dyn](#).
2. **Interfaz de selección:** Una vez que el script esté en ejecución, sigue los mismos pasos que se indican en los scripts [Conductos i.dyn](#) y [Conductos I.dyn](#). Deberás seleccionar en orden secuencial la ruta del enrutado por el suelo.
3. **Configuración del diámetro del tubo:** Al igual que en los scripts anteriores, configura el diámetro del tubo de acuerdo con las unidades configuradas en Revit.
4. **Presiona el botón “Entubar”:** Una vez que hayas seleccionado la ruta y configurado el diámetro del tubo, presiona el botón “Entubar” en la interfaz del script.

Siguiendo estos pasos, podrás emplazar enrutados de circuitos por el suelo en tu proyecto de manera eficiente. Asegúrate de personalizar el diámetro del tubo según las necesidades de tu proyecto eléctrico.

#### 4.2.6. Script Cuadros de carga.dyn

Este script es una herramienta compleja diseñada para el cálculo de cuadros de carga. Para su uso eficiente, sigue estos pasos:

1. **Ejecución del script:** Carga el script [Cuadros de carga.dyn](#). Ten en cuenta que este script puede ser más pesado en términos de recursos, por lo que se recomienda no tener muchas ventanas abiertas en tu software para evitar problemas en la ejecución.
2. **Define la base de la sección 050 del CNE-U:** Asegúrate de estar seguro sobre la base de la **Sección 050** del *Código Nacional de Electricidad-Utilización* que utilizarás para el cálculo de carga.
3. **Agrega información de áreas:** Incorpora la información de las áreas necesarias para el cálculo. Puedes obtener estos datos directamente del arquitecto proyectista o utilizando herramientas BIM en Revit para cuantificar las áreas.
4. **Agrega la carga adicional:** Incluye cualquier carga adicional que sea necesaria para el cálculo. Asegúrate de diferenciar claramente los distintos tipos de carga.
5. **Ejecuta el cálculo:** Después de configurar todas las variables necesarias, ejecuta el cálculo presionando el botón correspondiente en la interfaz del script.
6. **Opción de exportar a Excel (opcional):** Si deseas exportar los resultados del cálculo a un archivo Excel, asegúrate de seleccionar esta opción al iniciar el script y elige un archivo Excel en blanco para guardar los resultados.
7. **Uso de herramientas externas para importar resultados en Revit:** Para incluir los resultados en Revit, se recomienda no utilizar las tablas generadas por Revit, ya que estas son referenciales para mostrar los resultados. En su lugar, utiliza una herramienta como “DiRoots” para Revit, que te permitirá importar un archivo Excel de forma profesional, mejorando la representación en tus planos y cuadros de carga.

Este script es una herramienta potente que te ayudará a realizar cálculos precisos de cuadros de carga. Asegúrate de seguir los pasos mencionados para aprovechar al máximo sus capacidades.

#### 4.2.7. Script Caidas de tension.dyn

Este script tiene la capacidad de realizar cálculos representativos de caídas de tensión. Antes de comenzar, se recomienda crear una vista independiente en 3D en Revit para representar los resultados. Sigue estos pasos para utilizarlo:

1. **Ejecución del script:** Carga el script [Caidas de tension.dyn](#).
2. **Selecciona un tablero:** Selecciona el tablero para el cual deseas realizar el cálculo de caídas de tensión. Ten en cuenta que cada tablero tiene su circuito alimentador y circuitos derivados, y el cálculo se realiza en función de esta diferenciación.
3. **Mapeo de enrutados de tubos y circuitos eléctricos:** Selecciona, en orden, primero el circuito alimentador y luego los circuitos derivados en orden (primero C1, luego C2 y así sucesivamente). Asegúrate de seleccionar los tubos asociados, no los circuitos representados en Revit.

4. **Ejecuta el cálculo:** Una vez que hayas mapeado los enrutados y los circuitos eléctricos, ejecuta el cálculo presionando el botón correspondiente en la interfaz del script.
5. **Representación en vista independiente en 3D:** Crea una vista independiente en 3D en Revit para representar los resultados del cálculo de caídas de tensión.
6. **Opción de exportar a Excel (opcional):** Al igual que en el script de Cuadros de carga, tienes la opción de exportar los resultados a un archivo Excel para su manipulación y posterior importación en Revit.

Este script es una herramienta valiosa para calcular y visualizar las caídas de tensión en tu proyecto eléctrico. Asegúrate de seguir los pasos mencionados para obtener resultados precisos y aprovechar al máximo sus capacidades.

## 5. Scripts Custom

La capacidad de que los scripts sean de código abierto es una ventaja importante, ya que permite a los usuarios personalizarlos y mejorarlos según sus necesidades específicas. También fomenta la colaboración y la comunidad de usuarios para compartir conocimientos y desarrollos adicionales.

Es fundamental leer y comprender la política que se establece en la primera sección del manual, ya que esto ayudará a garantizar un uso adecuado de los scripts y evitar problemas de licencia o de cumplimiento con las normativas. Además, puede proporcionar orientación sobre cómo contribuir y mejorar la comunidad de usuarios.

Si deseas realizar modificaciones a los scripts existentes o desarrollar nuevos scripts basados en los algoritmos proporcionados, asegúrate de comprender completamente el funcionamiento de los scripts originales y de respetar las políticas y licencias asociadas. La colaboración y el intercambio de conocimientos son fundamentales para aprovechar al máximo las capacidades de código abierto.

Si tienes preguntas específicas o necesitas orientación sobre cómo personalizar los scripts o colaborar en su mejora, no dudes en preguntar. Puedes comunicarte con el desarrollador a través del correo [150153@unsaac.edu.pe](mailto:150153@unsaac.edu.pe).

### 5.1. ¿Cómo editar los scripts?

Editar los scripts es un proceso relativamente sencillo. Aquí tienes una guía básica sobre cómo hacerlo:

1. Inicia Revit y la plataforma de Dynamo. Asegúrate de tener Dynamo instalado y listo para su uso.
2. Carga el script que deseas editar o personalizar. Puedes hacerlo seleccionando “Archivo” en Dynamo y luego “Abrir”, y eligiendo el script que desees modificar.
3. **Explora el script:** Una vez que el script esté cargado, podrás ver los nodos y conexiones. Los nodos están agrupados y organizados en grupos lógicos.
4. **Comentarios y nodos Python:** Dentro del script, es posible que encuentres comentarios que explican la función de ciertos nodos o secciones. Además, podrás ver nodos Python que pueden tener líneas de código comentadas para ayudarte a comprender su funcionamiento.
5. **Tesis como referencia:** Como se mencionó en el manual, puedes consultar la tesis que originó esta propuesta para obtener una comprensión más profunda de los procesos algorítmicos que respaldan el desarrollo de estas herramientas. Esto puede ser especialmente útil si deseas realizar modificaciones avanzadas o personalizaciones en los scripts.

Ten en cuenta que al editar los scripts, debes tener conocimientos en programación y comprender los principios de diseño eléctrico en Revit. Además, es importante respetar las políticas de licencia y uso que se han establecido para estos scripts, especialmente si planeas compartir tus modificaciones o desarrollos adicionales con otros usuarios.

La edición de scripts en Dynamo ofrece una gran flexibilidad para personalizar y mejorar las capacidades de las herramientas, lo que puede ser muy beneficioso para adaptarlas a tus necesidades específicas.

## 6. Preguntas Frecuentes

1. **¿La propuesta puede ser emigrada a otros software BIM?**  
A la fecha, estos scripts tienen solo compatibilidad con Revit 2022.
2. **¿No me aparece la actualización de paquetes faltantes?**  
Para solucionar esto te recomendamos abrir cada script, verificar el nombre de cada nodo y buscar una referencia en internet, podrás encontrar fácilmente información sobre el paquete faltante e instalarlo de forma manual.
3. **¿Puedo ejecutar directamente desde Dynamo y no desde el reproductor?**  
Sí. Sin embargo podrías tener problemas de bucles infinitos en las interfaces, en ese caso, recuerda configurar el corrimiento de los scripts en “Manual” más no “Automático”.
4. **¿Dialux tiene problemas al importar el IFC?**  
Los problemas con el ifc en Dialux son comunes ya que en algunas versiones de este, se encuentra en fase beta. En ese caso, te recomendamos modelar un archivo BIM con información básica como muros, techos, suelos y aberturas.
5. **¿Los scripts Python no funcionan?**  
Es probable que requieras reinstalar la versión de Python más reciente o la que te preste más funcionalidad en tu PC en general.
6. **¿Los scripts pueden venderse a terceros?**  
No. Su desarrollo ha sido con puros fines académicos, por lo cual se insta a los interesados en lucrar con la propuesta a abstenerse, ya que esta configura propiedad intelectual del autor.
7. **¿Puedo subir los scripts a algún repositorio web?**  
Sí. No obstante recuerda dar los créditos necesarios al autor.
8. **¿Funcionarán los scripts en versiones superiores de Revit?**  
Probablemente sí. Las versiones de Dynamo al día de hoy normalmente mejoran en su interfaz visual y la forma de operación con paquetes de terceros, por lo cual, el inconveniente sería con la paquetería externa, en ese escenario se recomienda mantener actualizados los paquetes y leer la documentación provista por los desarrolladores correspondientes.
9. **Algunas de las soluciones propuestas no funcionan ¿Puedo contactar al autor?**  
Claro que sí. El correo de contacto del autor es [150153@unsaac.edu.pe](mailto:150153@unsaac.edu.pe).

## 7. Acerca del desarrollador

Estos scripts han sido desarrollados por Luis Giuseppe Yucra Altamirano como parte de su propuesta académica para obtener el título de ingeniero eléctrico en la Universidad Nacional de San Antonio Abad del Cusco en el año 2023.

Luis Giuseppe Yucra Altamirano es un apasionado ingeniero electricista originario de Cusco, Perú. Ha trabajado en diversos proyectos relacionados con el diseño eléctrico, lo que ha enriquecido su conjunto de habilidades y su comprensión del campo.

Su motivación para desarrollar estos scripts radica en la constante mejora del sector construcción en el ámbito de la ingeniería eléctrica y en la creencia en la importancia de la innovación en el diseño eléctrico. Su objetivo a futuro es lograr una implementación BIM mucho más eficiente en el desarrollo de la especialidad de instalaciones eléctricas.

Esta propuesta, respaldada por una sólida formación, experiencia y pasión, refleja el compromiso con la mejora continua en el campo de la ingeniería eléctrica y el diseño. Los scripts aquí presentados son una manifestación de habilidad y dedicación en este campo.

También está presto a recibir consultas, sugerencias o reporte de errores; el medio de comunicación es a través del siguiente correo [150153@unsaac.edu.pe](mailto:150153@unsaac.edu.pe).