



UNIVERSIDADE FEDERAL DE PERNAMBUCO  
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA  
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

ÍTALO RICARDO ALVES SILVA

**APLICAÇÃO DA METODOLOGIA *BUILDING INFORMATION MODELING* (BIM) NA  
EXECUÇÃO DE PROJETOS FOTOVOLTAICOS ASSISTIDA POR AUTOMAÇÕES  
UTILIZANDO O *SOFTWARE DYNAMO***

Recife  
2023

ÍTALO RICARDO ALVES SILVA

**APLICAÇÃO DA METODOLOGIA *BUILDING INFORMATION MODELING* (BIM)  
NA EXECUÇÃO DE PROJETOS FOTOVOLTAICOS ASSISTIDA POR  
AUTOMAÇÕES UTILIZANDO O *SOFTWARE DYNAMO***

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Elétrica da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Orientador(a): Prof. Dr. Gustavo Medeiros de Souza Azevedo

Recife  
2023

Ficha de identificação da obra elaborada pelo autor,  
através do programa de geração automática do SIB/UFPE

Silva, Ítalo Ricardo Alves.

Aplicação da metodologia Building Information Modeling (BIM) na execução de projetos fotovoltaicos assistida por automações utilizando o software Dynamo / Ítalo Ricardo Alves Silva. - Recife, 2023.

75 : il., tab.

Orientador(a): Gustavo Medeiros de Souza Azevedo

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, Engenharia Elétrica - Bacharelado, 2023.

Inclui referências, apêndices.

1. Projeto de plantas fotovoltaicas. 2. Building Information Modeling (BIM). 3. Revit. 4. Dynamo. 5. Python. I. Azevedo, Gustavo Medeiros de Souza. (Orientação). II. Título.

620 CDD (22.ed.)

ÍTALO RICARDO ALVES SILVA

**APLICAÇÃO DA METODOLOGIA *BUILDING INFORMATION MODELING* (BIM)  
NA EXECUÇÃO DE PROJETOS FOTOVOLTAICOS ASSISTIDA POR  
AUTOMAÇÕES UTILIZANDO O SOFTWARE DYNAMO**

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Elétrica da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Aprovado em: 28/09/2023.

**BANCA EXAMINADORA**

---

Prof. Dr. Gustavo Medeiros de Souza Azevedo (Orientador)  
Universidade Federal de Pernambuco

---

Prof. Dr. Fabrício Bradaschia (Examinador Interno)  
Universidade Federal de Pernambuco

---

Prof. Dr. Rafael Cavalcanti Neto (Examinador Interno)  
Universidade Federal de Pernambuco

Este trabalho é dedicado a todos que contribuíram para sua construção; Deus, meus pais, minha irmã, familiares e amigos, que sempre me incentivaram a prosperar por intermédio da engenharia

## **AGRADECIMENTOS**

Quero agradecer primeiramente a Deus, por todas as graças em minha vida, por ter me encorajado em todos os desafios apresentados e por ter me guiado em minha trajetória até a formação, sempre me instruindo pelo caminho do bem.

Deixo expressa aqui também minha mais sincera gratidão à minha família. Minha mãe, Edileuza Pacheco, com quem sempre pude contar e que cuida de mim de todas as formas. Minha irmãzinha, Maria Isabelly, que sempre me estimula ser uma pessoa melhor, mesmo quando não tem a intenção. Meu amado pai, Jeová Silva, que foi quem mais acreditou no meu potencial, quem me incentivou e deu todo o suporte para que eu conquiste meus sonhos, enfrentando qualquer desafio; ele que sempre foi meu amigo e professor, é também responsável por todas as conquistas em minha vida; a ele devo tudo, te amo eternamente. Sou grato também aos meus avós, tias e tio que, até nas pequenas coisas, sempre me ajudaram muito, me impulsionando na caminhada da graduação.

Agradeço também aos meus amigos e amiga conterrâneos, e minha namorada, que enfrentaram juntos comigo esses anos de formação, pois, apesar da distância necessária, sempre se fizeram presentes. Sou muito grato pelas pessoas que conheci nessa jornada e que se tornaram muito queridos. Amigos e amigas da universidade, com quem dividi muitas horas de estudo e momentos de descontração, e amigos do trabalho, que se tornaram companheiros da vida.

O homem científico não almeja resultados imediatos. Ele não espera que suas ideias mais avançadas sejam rapidamente retomadas. Seu trabalho é como o de um agricultor para o futuro. (NIKOLA TESLA).

## RESUMO

Nos últimos anos, a energia solar fotovoltaica tem apresentado crescimento exponencial em quantidade de potência instalada em todo o mundo, se tornando uma tecnologia cada vez mais avançada, confiável e acessível. Esse cenário é resultado, além de outros fatores, do aumento no número de empresas que surgiram para atuar em um segmento que se encontra bastante aquecido. Contudo, apesar de ser um mercado promissor, a parte técnica que envolve a elaboração dos projetos das usinas solares ainda não se encontra tão desenvolvida quanto deveria devido à baixa adoção às tecnologias mais avançadas, como as provenientes do uso da metodologia *Building Information Modeling* (BIM). Nesse contexto, pretende-se com esse trabalho discorrer sobre as implicações do BIM e como ele já é aplicado na engenharia, além de apresentar procedimentos e metodologias a serem executadas para o desenvolvimento de modelos paramétricos no *software* Revit, visto que esses elementos inteligentes é que são utilizados nos projetos elétricos. Outro objetivo principal é a exploração de um ambiente mais inovador para o setor da engenharia elétrica, com a otimização do trabalho utilizando automações no *software* Dynamo para realizar atividades e cálculos necessários que seriam feitos manualmente.

**Palavras-chave:** Projeto de plantas fotovoltaicas; *Building Information Modeling* (BIM); Revit; Dynamo; Python.

## ABSTRACT

In recent years, photovoltaic solar energy has shown exponential growth in installed capacity worldwide, becoming an increasingly advanced, reliable, and affordable technology. This scenario is the result, among other factors, of the increase in the number of companies that have sprung up to operate in a segment that is quite heated. However, despite being a promising market, the technical side of solar power plant design is still not as developed as it should be due to the low adoption of more advanced technologies, such as those resulting from using the Building Information Modeling (BIM) methodology. In this context, this work aims to discuss the implications of BIM and how it is already applied in engineering, as well as present procedures and methodologies to be carried out for the parametric model development in Revit software since these intelligent elements are used in electrical projects. Another main objective is to explore a more innovative environment for the electrical engineering sector, optimizing work by using automation in Dynamo to carry out necessary activities and calculations that would otherwise be done manually.

**Keywords:** Photovoltaic plant projects; Building Information Modeling (BIM); Revit; Dynamo; Python.

## LISTA DE ILUSTRAÇÕES

|   |    |
|---|----|
| Figura 1 – Metodologia usada para execução de projetos nas principais construtoras e incorporadoras brasileiras em 2019. .... | 17 |
| Figura 2 – Diferenças entre o processo tradicional e processos BIM para construção de um projeto. ....                        | 21 |
| Figura 3 – Plano genérico para implementação da metodologia BIM. ....   | 22 |
| Figura 4 – Evolução da potência total instalada da fonte solar fotovoltaica no Brasil até junho de 2023. ....                 | 25 |
| Figura 5 – Interface inicial do Revit 2023. ....  | 29 |
| Figura 6 – Interface de trabalho padrão no Revit 2023 para projetos elétricos. ....   | 30 |
| Figura 7 – Tipos de categorias para famílias criadas no Revit 2023. ....  | 31 |
| Figura 8 – Acessando o Dynamo e sua interface de usuário via Revit 2023. ....   | 33 |
| Figura 9 – Interface principal do Dynamo. ....  | 34 |
| Figura 10 – Exemplo genérico de bloco do Dynamo. ....   | 35 |
| Figura 11 – Algoritmo básico no Dynamo para consulta da parâmetros elétricos. ....  | 35 |
| Figura 12 – Modelo padrão para bloco de programação em Python no Dynamo. ....   | 36 |
| Figura 13 – Controle de unidades do projeto no Revit. ....  | 38 |
| Figura 14 – Criando sólidos para construção de famílias. ....   | 39 |
| Figura 15 – Extrusão vazia dando forma a geometria complexa. ....   | 40 |
| Figura 16 – Família Revit de inversor Sungrow para modelos SG40CX e SG50CX. ....  | 40 |
| Figura 17 – Criação de parâmetros de família no Revit. ....   | 41 |
| Figura 18 – Família paramétrica de módulos com tipos para diversos modelos. ....  | 43 |
| Figura 19 – Principais famílias criadas para projetos e automações. ....  | 44 |
| Figura 20 – Demonstração do uso de condições em parâmetros de família. ....   | 46 |
| Figura 21 – Controle do diagrama trifilar de uma MPPT com três <i>strings</i> . ....  | 47 |
| Figura 20 – Construção e controle do diagrama trifilar de inversor fotovoltaico. ....   | 48 |
| Figura 21 – Associação de parâmetros entre famílias aninhadas. ....   | 48 |
| Figura 24 – Famílias paramétricas de diagramas trifilares para projetos. ....   | 49 |
| Figura 25 – Projeto civil vinculado ao projeto elétrico já com alguns equipamentos. ....                                      | 50 |
| Figura 26 – Visão geral da rotina para dimensionamento e montagem do quadro de proteção. ....                                 | 51 |
| Figura 27 – <i>Script</i> para dimensionamento do disjuntor e dos DPS. ....   | 52 |
| Figura 28 – Resultado gerado pela rotina de montagem do quadro de proteção. ....  | 53 |
| Figura 29 – Fluxograma do <i>script</i> para a automação de montagem dos quadros de proteção. ....                            | 54 |
| Figura 29 – Visão geral da rotina de criação de diagramas trifilares. ....  | 55 |
| Figura 30 – Fluxograma e <i>script</i> para criação de diagrama trifilar do inversor. ....                                    | 56 |
| Figura 31 – <i>Script</i> para separação das strings nas MPPTs do inversor. ....  | 57 |
| Figura 32 – Resultado gerado pela rotina de criação de diagramas trifilares. ....   | 58 |
| Figura 31 – Fluxograma do <i>script</i> para a automação de criação de diagramas trifilares. ....                             | 59 |
| Figura 33 – <i>Script</i> completo e resultados para busca de instância do inversor FV. ...                                   | 63 |
| Figura 34 – <i>Script</i> completo e resultados para busca de pontos do inversor. ....  | 63 |
| Figura 35 – <i>Script</i> completo e resultados para definição de vetores do inversor. ....                                   | 64 |
| Figura 36 – <i>Script</i> completo e resultados para filtragem de equipamentos elétricos. ....                                | 64 |
| Figura 37 – <i>Script</i> completo e resultados para inserir instância de quadro metálico. ....                               | 65 |
| Figura 38 – <i>Script</i> com dados de cabo e seu respectivo disjuntor. ....  | 65 |
| Figura 39 – <i>Script</i> completo e resultados para filtragem de dispositivos elétricos. ...                                 | 66 |

|   |    |
|---|----|
| Figura 40 – <i>Script</i> completo e resultados para dimensionamento de dispositivos. ...                             | 66 |
| Figura 41 – <i>Script</i> completo para parametrização dos dispositivos. ....   | 67 |
| Figura 42 – <i>Script</i> completo e resultados para busca de equipamentos e circuitos elétricos. ....                | 68 |
| Figura 43 – Bloco que separa elementos dos circuitos nos painéis elétricos. ....                                      | 68 |
| Figura 44 – Blocos para busca de todos os tipos da categoria Modelos genéricos. .                                     | 69 |
| Figura 45 – Bloco para seleção da caixa de escopo. ....   | 69 |
| Figura 46 – <i>Script</i> completo e resultados para determinação dos vértices da caixa de escopo. ....               | 70 |
| Figura 47 – <i>Script</i> completo e resultados determinação da origem do diagrama trifilar. ....                     | 70 |
| Figura 48 – <i>Script</i> completo e resultados para determinação de vetores da caixa de escopo. ....                 | 71 |
| Figura 49 – Bloco para a organização dos fios de comunicação entre blocos. ....                                       | 71 |
| Figura 50 – <i>Script</i> completo e resultados para busca de todos os parâmetros do inversor. ....                   | 72 |
| Figura 51 – <i>Script</i> completo e resultados para filtrar parâmetros desejados. ....                               | 72 |
| Figura 52 – <i>Script</i> completo e resultados para separação dos parâmetros dos dispositivos. ....                  | 73 |
| Figura 53 – <i>Script</i> completo e resultados para inserir instância de cabos entre inversor e QPG. ....            | 73 |
| Figura 54 – <i>Script</i> completo e resultados para inserir instância de quadro metálico. .                          | 74 |
| Figura 55 – <i>Script</i> completo e resultados para tratar as informações das <i>strings</i> . ....                  | 74 |
| Figura 56 – <i>Script</i> completo e resultados para a busca de vetores normais aos módulos FV. ....                  | 75 |
| Figura 57 – <i>Script</i> completo e resultados para criar diagramas trifilares de <i>strings</i> em suas MPPTs. .... | 75 |

## LISTA DE TABELAS

|  |    |
|--|----|
| Tabela 1 – Fases de implementação do BIM definidas no pelo decreto nº 10.306. ...                | 24 |
| Tabela 2 – Descrição de parâmetros da família de inversores Sungrow modelos SG40CX e SG50CX..... | 42 |
| Tabela 3 – Operações aritméticas utilizadas nas fórmulas dos parâmetros das famílias. ....       | 45 |
| Tabela 4 – Condições e operações lógicas utilizadas em fórmulas dos parâmetros das famílias..... | 45 |
| Tabela 5 – Características do sistema solar fotovoltaico do caso em estudo. ....                 | 50 |

## LISTA DE ABREVIATURAS E SIGLAS

|      |  |
|------|--|
| AEC  | Arquitetura, Engenharia e Construção   |
| API  | <i>Application Programming Interface</i> – Interface de Programação de Aplicação |
| BIM  | <i>Building Information Modeling</i> – Modelagem da Informação da Construção     |
| BT   | Baixa Tensão   |
| CA   | Corrente Alternada   |
| CAD  | <i>Computer Aided Design</i> – Desenho Assistido por Computador                  |
| DPS  | Dispositivo de Proteção Contra Surtos  |
| FV   | Fotovoltaico   |
| IFC  | <i>Industry Foundation Classes</i>   |
| MPPT | <i>Maximum Power Point Tracking</i> – Seguimento do Ponto de Máxima Potência     |
| SAP  | <i>System Applications and Products</i>  |
| SQL  | <i>Structured Query Language</i> – Linguagem de Consulta Estruturada             |
| UFV  | Usina Solar Fotovoltaica   |
| 2D   | Bidimensional  |
| 3D   | Tridimensional   |

## LISTA DE SÍMBOLOS

|      |   |
|------|---|
| .RVT | Extensão para arquivo de projeto do Revit         |
| .RFA | Extensão para arquivo de família do Revit         |
| .RTE | Extensão para arquivo de <i>template</i> do Revit |
| ID   | <i>Idendity</i>                                   |
| IN   | <i>Input</i>                                      |
| OUT  | <i>Output</i>                                     |
| Voc  | Tensão de circuito aberto                         |
| Isc  | Corrente de curto-circuito                        |
| Pmax | Potência máxima                                   |

## SUMÁRIO

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>INTRODUÇÃO</b> .....                                | <b>15</b> |
| 1.1      | OBJETIVOS .....  | 18        |
| 1.1.1    | Geral .....  | 18        |
| 1.1.2    | Específicos .....                                      | 18        |
| 1.2      | ORGANIZAÇÃO DO TRABALHO .....                          | 19        |
| <b>2</b> | <b>FUNDAMENTAÇÃO TEÓRICA</b> .....                     | <b>20</b> |
| 2.1      | CONCEITO BIM .....                                     | 20        |
| 2.1.1    | Interoperabilidade .....                               | 22        |
| 2.2      | APLICABILIDADE E IMPLEMENTAÇÃO .....                   | 23        |
| 2.2.1    | Cenário brasileiro quanto à implementação do BIM ..... | 24        |
| 2.2.2    | Aplicação em projetos fotovoltaicos .....              | 25        |
| 2.3      | SOFTWARES UTILIZADOS .....                             | 26        |
| 2.4      | ENQUADRAMENTO METODOLÓGICO .....                       | 27        |
| 2.5      | VISÃO GERAL E PRINCIPAIS FERRAMENTAS DO REVIT .....    | 28        |
| 2.5.1    | Famílias no Revit .....                                | 30        |
| 2.5.1.1  | <i>Parâmetros e suas propriedades</i> .....            | 32        |
| 2.6      | INTERFACES DE PROGRAMAÇÃO NO DYNAMO .....              | 32        |
| 2.6.1    | Visão geral do ambiente .....                          | 32        |
| 2.6.2    | Definições e operação com o Dynamo .....               | 34        |
| 2.6.3    | Python e API do Revit .....                            | 36        |
| <b>3</b> | <b>FAMÍLIAS REVIT E AUTOMAÇÕES NO DYNAMO</b> .....     | <b>37</b> |
| 3.1      | OBJETOS PARA DESENVOLVIMENTO .....                     | 37        |
| 3.2      | CONSTRUÇÃO DAS FAMÍLIAS PARAMETRIZADAS .....           | 37        |
| 3.2.1    | Família de inversor .....                              | 37        |
| 3.2.2    | Família de diagrama trifilar do inversor .....         | 45        |
| 3.3      | ROTINA PARA DIMENSIONAMENTO DO QUADRO ELÉTRICO .....   | 51        |
| 3.4      | ROTINA PARA CRIAÇÃO DE DIAGRAMAS TRIFILARES .....      | 54        |
| <b>4</b> | <b>CONCLUSÕES E PROPOSTAS DE CONTINUIDADE</b> .....    | <b>60</b> |
|          | <b>REFERÊNCIAS</b> .....                               | <b>61</b> |
|          | <b>APÊNDICES</b> .....                                 | <b>63</b> |

## 1 INTRODUÇÃO

A elaboração de um projeto elétrico consiste em realizar os cálculos para o dimensionamento dos dispositivos e elementos a serem instalados em determinado empreendimento, seguindo as normativas aplicáveis. Conforme Creder [1], o projeto de instalações elétricas “é a previsão escrita da instalação, com todos os seus detalhes, a localização dos pontos de utilização da energia elétrica, os comandos, os trajetos dos condutores, a divisão em circuitos, a seção dos condutores, os dispositivos de manobra, a carga de cada circuito, a carga total etc.”.

Sendo assim, faz parte do escopo de projeto a representação gráfica e descritiva de todo o sistema, para que o mesmo seja implantado corretamente e para que esse material seja uma boa referência em manutenções futuras. Nesse sentido, a execução bem elaborada de projetos elétricos carrega a responsabilidade do funcionamento adequado de sistemas que estarão em atividade por décadas e que atenderão as necessidades de vários indivíduos, pois tornaram-se fundamentais à convivência humana, ao passo que a sociedade vem aumentando o consumo da energia elétrica, ficando cada vez mais dependente dela [2].

Os serviços de Arquitetura, Engenharia e Construção (AEC) precedem a existência da indústria moderna de hoje. Dessa forma, os procedimentos e métodos usados para a elaboração de projetos acompanharam a evolução dos meios e da tecnologia com o passar dos anos [3], iniciando com o desenho manual, utilizando papel, lápis, régua e demais acessórios. Nesse cenário, era demandado bastante tempo laboral do projetista para obter-se resultados precisos, processo esse que se tornava extenuante com atividades repetitivas a depender do tamanho do projeto. Além disso, pequenas modificações ou correções poderiam ser motivo de refazer o trabalho como um todo.

O primeiro avanço no setor veio com a integração do Desenho Assistido por Computador (em inglês, *Computer Aided Design* – CAD), que acelerou e deu precisão no desenvolvimento gráfico com linhas, curvas, blocos inseridos precisamente e, além disso, integrou legendas e tabelas com cálculos diretamente no arquivo, dentre outras funcionalidades [4]. Ainda que a dinâmica de elaboração de projetos tenha mudado, ficando mais rápida, prática e precisa, se conservavam as etapas de atividades realizadas manualmente.

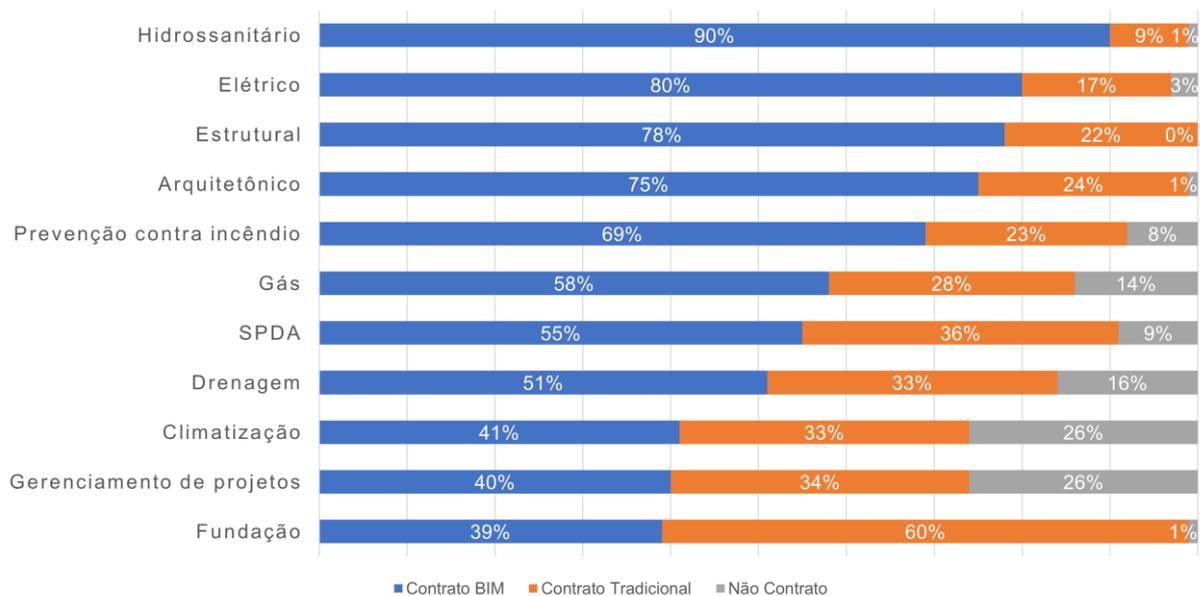
Nesse contexto, a proposta mais atual e inovadora do mercado que atende a essa problemática é a metodologia da modelagem da informação da construção (em inglês, *Building Information Modeling – BIM*), que, quando aplicada corretamente, proporciona tudo que o CAD oferece e integra diversas ferramentas, permitindo que diferentes áreas de atuação envolvidas no desenvolvimento de um mesmo projeto trabalhem simultaneamente. Isso é possível pois o BIM dispõe de recursos que permitem uma elevada análise e controle do projeto, desde que os modelos virtuais sejam criados com equivalente precisão [5].

A adesão à tecnologia BIM promove avanços expressivos no mercado AEC, marcando também o envolvimento íntimo com um dos setores mais promissores do mundo moderno, o de programação, que, quando aplicado no mercado de engenharia, tem o potencial de proporcionar resultados mais precisos e satisfatórios, tanto para a empresa responsável quanto para o cliente final e seus usuários. Apesar das vantagens e benefícios resultantes do uso do BIM serem entendidos pela maioria dos profissionais da área, ainda há resistência quanto à implantação da metodologia nos setores das empresas. Em contrapartida, grandes instituições historicamente vêm investindo em estudos para o estabelecimento dessa tecnologia. Dentro dessas circunstâncias, a biblioteca digital e o progresso tecnológico para mercado de engenharia elétrica, e principalmente de energia solar fotovoltaica, apresentam-se pouco desenvolvidos se comparados aos resultados já alcançados em trabalhos civis e arquitetônicos [6,7], por exemplo.

Essa característica aponta uma oportunidade promissora para o setor elétrico com enorme potencial para otimizar a execução de projetos e aprimorar a consistência do produto obtido, pois ambos os resultados são alcançados através da implementação e criação de alternativas com ferramentas BIM. Um estudo anual é realizado pela empresa especializada em plataformas inteligentes, Thórus Engenharia, denominado por Cenário Construtivo Brasileiro [8] e que tem como objetivo comparar as operações das principais construtoras e incorporadoras em atuação no Brasil. A versão de 2019 da pesquisa constatou que apenas 48% das 167 empresas participantes do estudo já trabalharam com BIM. Nesse montante de cerca de 80 companhias, consultou-se qual metodologia é adotada para a execução de cada projeto nas diferentes áreas da engenharia atendidas por cada empresa. A Figura 1 apresenta o resultado dessa pesquisa, onde é possível constatar que, nesse cenário,

80% dos contratos para projetos elétricos são realizados utilizando tecnologias BIM, enquanto 17% ainda seguem os procedimentos tradicionais. Para mercados mais desenvolvidos com a metodologia, como o estrutural e o arquitetônico, a adoção do BIM foi menor, 78% e 75%, respectivamente. Essas características ressaltam dois pontos importantes, onde o primeiro deles é que as empresas que já executaram projetos em BIM tendem a expandir sua aplicação nos demais contratos, pois, certamente, colheram benefícios que justificam sua continuação. O segundo ponto é que, apesar de historicamente o BIM ter prosperado mais em outras áreas da engenharia, o setor elétrico enxerga solidez na escolha do BIM para a contratação dos projetos, mesmo que, comparativamente, com poucos recursos disponíveis.

Figura 1 – Metodologia usada para execução de projetos nas principais construtoras e incorporadoras brasileiras em 2019.



Fonte: adaptado de [8].

As tecnologias BIM fornecem as ferramentas mais indicadas quando se pretende otimizar o processo construtivo dos projetos elétricos, pois todos os dados necessários estão disponíveis para serem acessados no próprio projeto. Essa característica permite, além de todos os benefícios inerentes à metodologia, potencializar a execução dos projetos, tornando o processo mais rápido e consistente através do uso de automações que realizam cálculos e montagens baseadas nos

resultados, atividades essas que seriam executadas manualmente pelos projetistas. Essa vertente que usa de *scripts* para automatizar processos BIM é, conseqüentemente, ainda menos explorada que a própria metodologia, todavia, espera-se que sua aplicação se torne uma tendência no futuro, já que o uso da programação nas engenharias vem se tornando cada vez mais presente.

## **1.1 Objetivos**

### **1.1.1 Geral**

Desenvolver e analisar os resultados de automações criadas para auxiliar os projetistas nas etapas de dimensionamento e montagem dos projetos elétricos fotovoltaicos, desde a construção do ambiente fundamentado na usabilidade da metodologia BIM até a construção dos *scripts*.

### **1.1.2 Específicos**

Para alcançar o objetivo principal, pretende-se por meio do presente trabalho tratar os seguintes objetivos específicos:

- a. Apresentar as ferramentas de trabalho Revit e Dynamo, definir as principais características e configurações necessárias para o correto funcionamento do ambiente e suas aplicações;
- b. Construção de modelos virtuais paramétricos dos equipamentos e dispositivos elétricos instalados em usinas solares fotovoltaicas, para posterior consumo de dados pelos *scripts*;
- c. Criação nos modelos de parâmetros responsivos aos dados de entrada lidos do projeto, de modo que os tornem controláveis por *script*;
- d. Desenvolvimento dos códigos de programação usando *DesignScript*, Python e a API do Revit, para a realização automática das atividades.

## 1.2 Organização do Trabalho

O presente trabalho é construído em cima de quatro capítulos, sendo o primeiro deles responsável por apresentar uma introdução ao tema principal abordado e por indicar os objetivos almejados ao fim do projeto.

No segundo capítulo é discorrido sobre a fundamentação teórica que envolveu o surgimento e a evolução da metodologia BIM, além de demonstrar as principais ferramentas, configurações e funcionalidades dos softwares que serão utilizados.

Em seguida, no terceiro capítulo, são apresentadas metodologias práticas de desenvolvimento de modelos, levantados os resultados obtidos, isto é, os modelos inteligentes construídos para utilização dentro do ambiente BIM e os *scripts* desenvolvidos no Dynamo.

No último capítulo são apresentadas tanto conclusões acerca do projeto desenvolvido como um todo, enfatizando na relação básica entre início e fim do processo, quanto a descrição de propostas para continuidade ou sugestões de aprimoramento para trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 Conceito BIM

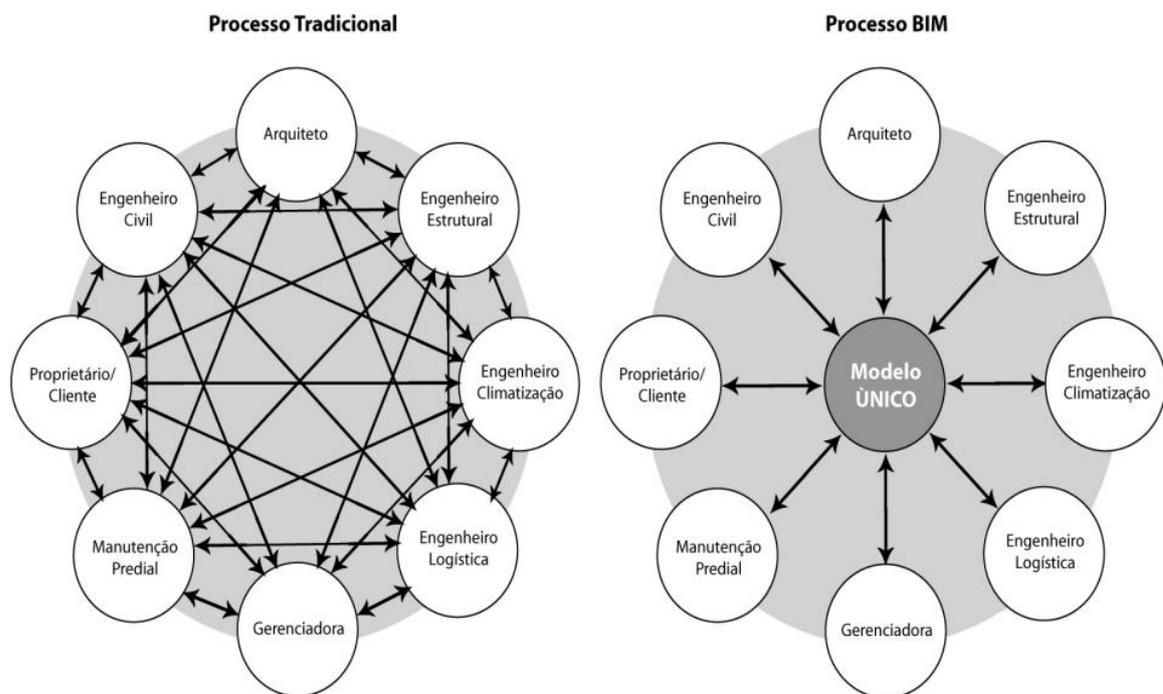
A modelagem da informação da construção (em inglês, *Building Information Modeling – BIM*) não é entendida de forma única, isto é, ela ainda não possui uma definição que seja amplamente aceita. No entanto, mesmo com as diversas significações recebidas, a ideia da construção de modelos virtuais sempre está presente, pois ela é peça fundamental para a operabilidade dessa ferramenta. De forma clara, então, pode-se entender como BIM a construção virtual de modelos reais, envolvendo um conjunto de políticas, processos e tecnologias interativas [9], com a finalidade de simular detalhadamente o ambiente construtivo, de modo que seja possível acompanhá-lo durante todo o seu ciclo de vida, facilitando o entendimento do projeto, auxiliando na prevenção contra erros e contribuindo para a tomada de decisões de forma assertiva.

Para tal propósito, a metodologia está respaldada na informação, como indica o próprio nome. Isto significa que, diferentemente de como é operado com ferramentas CAD, no BIM todo e qualquer modelo virtual tridimensional (3D) carrega parâmetros com representações que definem suas características reais e o distingue de todos os outros, possibilitando seu funcionamento e autonomia [10]. Para equipamentos e dispositivos elétricos, por exemplo, além das dimensões, que é o tipo de dado indispensável a qualquer modelo, é preciso incluir no objeto 3D características como tensão, corrente e potência nominais, de pico etc. Da mesma forma que, pressupondo a necessidade da análise de custo e desempenho de módulos fotovoltaicos, outras informações podem ser incluídas em suas representações, como fabricante, preço, validade, performance garantida e taxa de degradação anual. Isso significa que os elementos criados e compartilhados nesse ambiente devem conter dados relevantes e suficientes sobre os empreendimentos reais que eles representam, compondo assim, dentro de um único arquivo, um banco de dados relacional de informações que constituem um modelo “inteligente” que se adapta automaticamente aos seus componentes [9,10].

Nessas circunstâncias, fica claro que o BIM não se trata de um *software* específico e sim de um conjunto de procedimentos adotados para um objetivo final: a

construção de um projeto completo e unificado, de modo que atenda a diferentes áreas e que todos os intervenientes sejam capazes de acompanhar de forma precisa a influência das suas atividades e o progresso do empreendimento como um todo, permitindo o processo de retroalimentação ainda na fase de projetos, sem a necessidade de buscar informações externas para realização de análises ou estimativas. Essa diferença entre o processo tradicional de desenvolvimento de um projeto e o processo BIM está ilustrada na Figura 2, onde os caminhos indicados pelas setas indicam a troca de informações entre as diferentes áreas.

Figura 2 – Diferenças entre o processo tradicional e processos BIM para construção de um projeto.

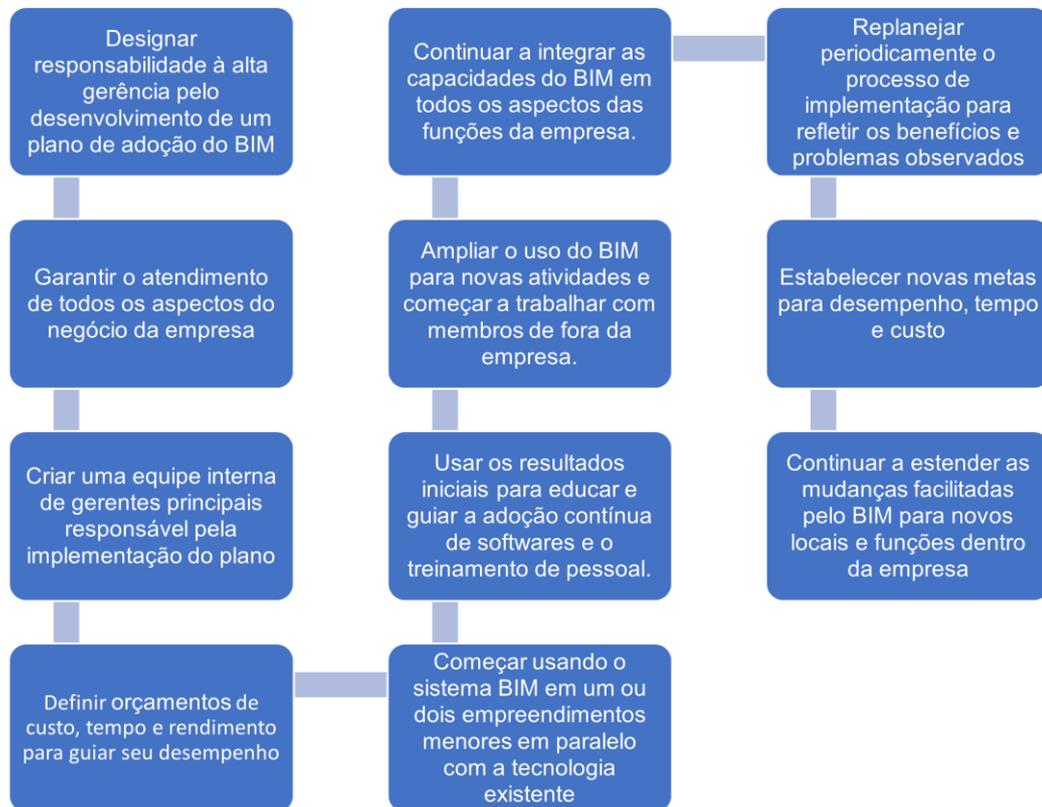


Fonte: extraído de [11]

Dessa forma, o BIM pode ser executado em diferentes plataformas, transitando de forma integrada entre elas com seus modelos, transmitindo confiabilidade proporcional a robustez das informações que carrega. Para Eastman [5], implantar efetivamente a metodologia requer aplicar mudanças necessárias em quase todos os aspectos do negócio da empresa e não somente fazer as mesmas coisas de uma nova maneira e esperar que a conversão aconteça. A Figura 3 mostra os passos

gerais de um dos diversos planos sugeridos para implantação da tecnologia em empresas.

Figura 3 – Plano genérico para implementação da metodologia BIM.



Fonte: adaptado de [5].

### 2.1.1 Interoperabilidade

Apesar de alguns *softwares* comportarem funcionalidades que atendam várias áreas de uma construção – como parte elétrica, hidráulica e civil, por exemplo, é comum os setores envolvidos utilizarem programas diferentes, cada um mais direcionado para seu contexto, atendendo especificamente às suas necessidades. Como uma das propostas do BIM é a unificação de todos os empreendimentos em um único modelo federado (pois nele é permitido alterações simultâneas), identifica-se então a necessidade de integrar as diferentes ferramentas sem que ocorra a perda ou duplicidade de informações. Essa capacidade de sistemas diferentes trocarem dados e os usarem é entendido como interoperabilidade [9]. Esse atributo elimina a

necessidade de replicar informações, geometrias e características já criados em outras etapas, facilitando os fluxos de trabalho e permitindo o uso de automações [5].

Para tal fim, é necessário um formato aberto de comunicação que todas as ferramentas BIM sejam capazes de interpretar e processar. O padrão mais utilizado para o mercado AEC, pois conta com um tipo de linguagem que foca na modelagem do produto e processos, é o *Industry Foundation Classes* (IFC), que se trata de um formato de arquivo com especificações padronizadas que permitem a análise do empreendimento desde etapa de planejamento até o gerenciamento pós-obra [7]. Esse formato de arquivo, quando gerado, carrega as propriedades dos objetos virtuais que representa, os materiais utilizados, a relação hierarquizada em categorias de todos os modelos, suas geometrias e localizações [5].

## **2.2 Aplicabilidade e implementação**

Mesmo diante da eficiência alcançada graças ao uso do BIM, para a realização de projetos e otimização de processos, a metodologia ainda não é tão utilizada quanto se espera, pois o cenário atual não é muito favorável quanto a sua implementação. Os desafios para a adoção dessa tecnologia são análogos ao de qualquer procedimento inovador. A primeira barreira é a financeira, pois é necessário alto investimento para custear melhorias de *hardware* e as licenças dos *softwares* que, a depender de quais sejam, oferecem planos mensais e/ou anuais, por exemplo. O segundo desafio é o fornecimento de treinamento para utilizar as ferramentas que, ao primeiro contato, não são simples e carecem de dedicação do operador para aperfeiçoar-se e começar a desempenhar bem. A última etapa, e a mais simples de ser superada quando o processo já está em curso, é a mudança de mentalidade dos envolvidos, estimulando-os a adotarem os novos procedimentos e a contribuírem para o melhoramento contínuo, tornando-os mais céticos quanto ao uso das técnicas tradicionais [12].

### 2.2.1 Cenário brasileiro quanto à implementação do BIM

À medida que uma tecnologia fica mais difundida o acesso a ela se torna mais simples, permitindo que seu desenvolvimento seja exponencial. Com a finalidade de promover um ambiente adequado ao investimento na metodologia, o governo brasileiro instituiu em 2018 a Estratégia Nacional de Disseminação do BIM, o BIM BR, através do decreto nº 9.377 [13], que foi revogado e atualizado em 2019 pelo decreto nº 9.983 [14]. Em 2020 foi publicado o decreto nº 10.306 que estabelece a utilização de forma gradual do BIM para a execução de obras e serviços de engenharia realizados pelos órgãos e pelas entidades públicas federais [15]. A implementação é dividida em fases e a descrição, bem como a data de vigência para cada uma delas, pode ser vista na Tabela 1. Ainda que tardiamente, se comparadas com o cenário internacional, essas medidas fomentaram a implementação das ferramentas nas empresas, o desenvolvimento da tecnologia e a disseminação do conhecimento sobre BIM no mercado nacional brasileiro.

Tabela 1 – Fases de implementação do BIM definidas no pelo decreto nº 10.306.

| Fase | Data de vigência      | Adequação  |
|------|-----------------------|--|
| I    | 1º de janeiro de 2021 | Desenvolvimento de projetos referentes a construções novas, ampliações ou reabilitações quando consideradas de grande relevância   |
| II   | 1º de janeiro de 2024 | Execução direta ou indireta de projetos e na gestão de obras, referentes a construções novas, reformas, ampliações ou reabilitações quando consideradas de grande relevância                         |
| III  | 1º de janeiro de 2028 | Desenvolvimento de projetos de arquitetura e engenharia e na gestão de obras referentes a construções novas, reformas, ampliações e reabilitações, quando consideradas de média ou grande relevância |

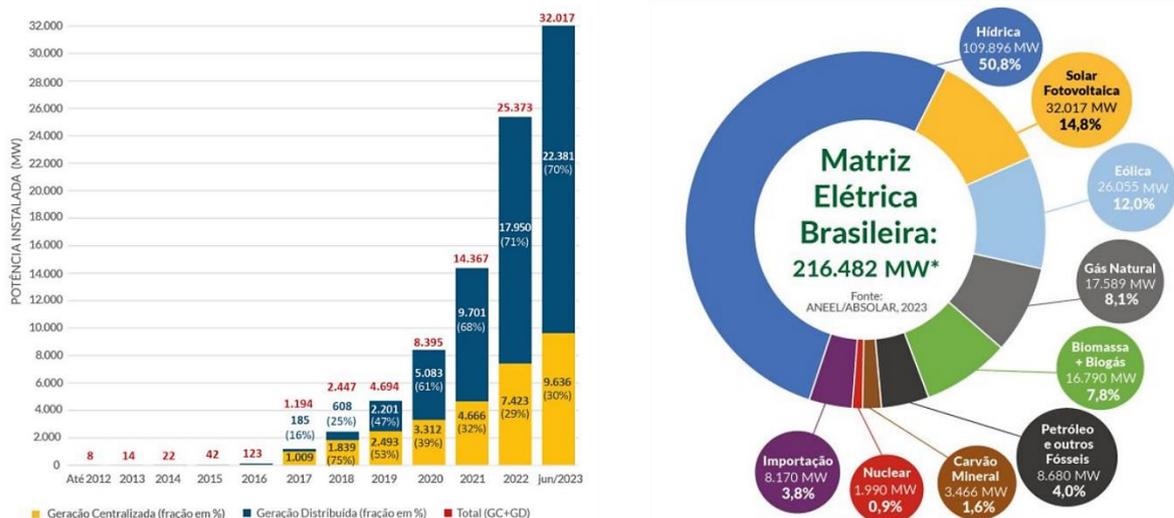
Fonte: adaptado de [15].

## 2.2.2 Aplicação em projetos fotovoltaicos

Nos últimos anos a energia solar fotovoltaica vem sendo um dos recursos mais estudados na área de geração de energia. Essa característica é consequência da elevada adoção desse modelo num cenário em que as corporações e os pequenos consumidores buscam reduzir os custos com o consumo de energia elétrica e, ao mesmo tempo, procuram minimizar os impactos ambientais. Além da disponibilidade abrangente e regular, características de uma fonte renovável, outra motivação para a adesão de sistemas fotovoltaicos é o baixo custo para a implantação de um empreendimento desse tipo quando comparado com os modelos convencionais de geração de energia, como as centrais hidráulicas e termoelétricas, por exemplo.

Consequentemente, essa acessibilidade é uma das qualidades que tem impulsionado o crescimento exponencial no número de implantações de pequenas usinas solares, principalmente em residências e pequenos comércios. No Brasil, o setor ultrapassou a marca de 32 GW de potência instalada em junho de 2023, com destaque para sistemas de microgeração (potência instalada  $\leq 75 \text{ kW}$ ) e minigeração ( $75 \text{ kW} \geq$  potência instalada  $\geq 3 \text{ MW}$ ) distribuída, responsáveis por 70% da categoria, como pode ser visto na Figura 4. Esses números consagram o setor de energia solar fotovoltaica como a segunda maior fonte geradora de energia do país, posição conquistada no início do primeiro semestre de 2023.

Figura 4 – Evolução da potência total instalada da fonte solar fotovoltaica no Brasil até junho de 2023.



Fonte: extraído de [16].

No que diz respeito a energia solar fotovoltaica no Brasil, apesar do país ser um dos mais promissores no setor, ainda há desafios a serem superados com respeito a melhorias que podem ser alcançadas por advento do uso da tecnologia na execução e no gerenciamento dos projetos. Esse cenário tem permitido uma exploração mais dedicada em duas das vertentes do BIM que serão mais abordadas no presente trabalho: a de fornecer celeridade na elaboração de projetos e a de garantir maior confiabilidade em seus resultados.

### 2.3 Softwares utilizados

As ferramentas BIM utilizadas como objeto de pesquisa no presente trabalho são o Revit e o Dynamo, ambos da Autodesk, empresa de *softwares* líder global em tecnologia de projetos voltados para o mercado AEC. A companhia também é proprietária do principal e ainda mais utilizado *software* CAD de todos os tempos, o AutoCAD. Apesar de algumas poucas semelhanças na operabilidade dessas ferramentas e na possibilidade de transferir arquivos entre elas, não há interoperabilidade entre ambas. Isso se deve ao fato de o Revit, como *software* BIM, funcionar como uma base de dados orientada a objetos enquanto o AutoCAD não suporta esse tipo de atividade, utilizando-se apenas das representações gráficas e carentes de informação do modelo virtual original. No Revit, esses dados estão presentes dentro dos próprios elementos 3D, diferentemente da maneira convencional onde são criados em tabelas ou listas e, a partir daí, são associadas analiticamente pelo operador aos modelos gráficos. Sendo assim, no Revit, o dimensionamento, a separação e a contabilização ocorrem de forma automática, pois nele os modelos virtuais é que preenchem e editam as tabelas, por exemplo.

O Dynamo é uma ferramenta de programação visual que, majoritariamente, é executado de forma semelhante a um *plug-in* no Revit. Assim sendo, as duas aplicações atuam bilateralmente, conversando perfeitamente entre si. Para tal, o Dynamo utiliza-se das linguagens de programação *DesignScript* e Python, onde a primeira é baseada na construção de diagramas de blocos e a segunda na descrição, sendo também especializada para análise de dados. Além disso, o *software* é capaz de executar diretamente a API do Revit, o que lhe confere autonomia para consumir dados e realizar quaisquer atividades no Revit que seriam executadas por um

operador humano. Sendo assim, juntas, essas duas ferramentas têm o potencial de realizar diversas atividades, das quais o Dynamo pode ser responsável por executar:

- a. construção de geometrias complexas, símbolos de anotação, textos e representações gráficas no Revit;
- b. leitura, criação, alteração, exclusão e transferência de dados;
- c. dimensionamento elétrico e verificação seguindo critérios (normas, por exemplo);
- d. inserção, seleção, edição e demais atividades com os modelos virtuais;
- e. operações com tabelas, criação de vistas e controle de revisões;
- f. criação e manuseio de arquivos externos;
- g. acesso e exportação para ferramentas externas como Excel, Word, PDF, bancos de dados relacionais com SQL, SAP, páginas web e APIs (a do Google Maps, por exemplo) dentre outras.

Essa gama de possibilidades abre oportunidade para o desenvolvimento de várias automações que auxiliariam na elaboração de projetos fotovoltaicos, eliminando diversos gargalos que atrasam a entrega como a necessidade de executar atividades repetitivas, ocorrência de retrabalhos, incidência de erros, dentre outros.

O Revit é um *software* de modelagem paramétrica que suporta nativamente diversos conjuntos de modelos das mais variadas classes e áreas da engenharia. Sendo assim, todo o ambiente será criado nele, para posteriormente ser manipulado pelo Dynamo. Como o objetivo é que todo esse ecossistema funcione de forma sincronizada, é preciso garantir três coisas fundamentais para o BIM: famílias parametrizadas corretamente, um *template* bem configurado e uma equipe bem treinada para operar seguindo a metodologia.

## 2.4 Enquadramento metodológico

Como o ambiente BIM ainda não é amplamente utilizado para o desenvolvimento de projetos solares fotovoltaicos, então os modelos virtuais de equipamentos e dispositivos elétricos, quando existem, não são facilmente encontrados. Sendo assim,

quando uma empresa que trabalha com projetos elétricos pretende implementar a metodologia, muitas vezes é preciso criar as representações que serão utilizadas em seus projetos. Nesse sentido, é importante, entender como construir os modelos virtuais, que no Revit são chamados de famílias, de acordo com as suas características reais, mas que sigam um plano de parametrização próprio definido pelo desenvolvedor. Esse grau de liberdade é poderoso, pois permite que as famílias respondam perfeitamente às necessidades específicas que o projeto exige e, além disso, possibilita a usabilidade esperada com as automações. Sendo este último o ponto de desenvolvimento principal do presente trabalho, é importante ressaltar que aplicar *scripts* para automatizar atividades em modelos BIM para elétrica é algo pouco explorado e as ferramentas disponíveis que fazem uso dessa metodologia possuem tecnologia fechada e não divulgada. Em contrapartida, para o mercado de engenharia civil diversos estudos são realizados para automatizar modelagens no Revit com o Dynamo [12,17].

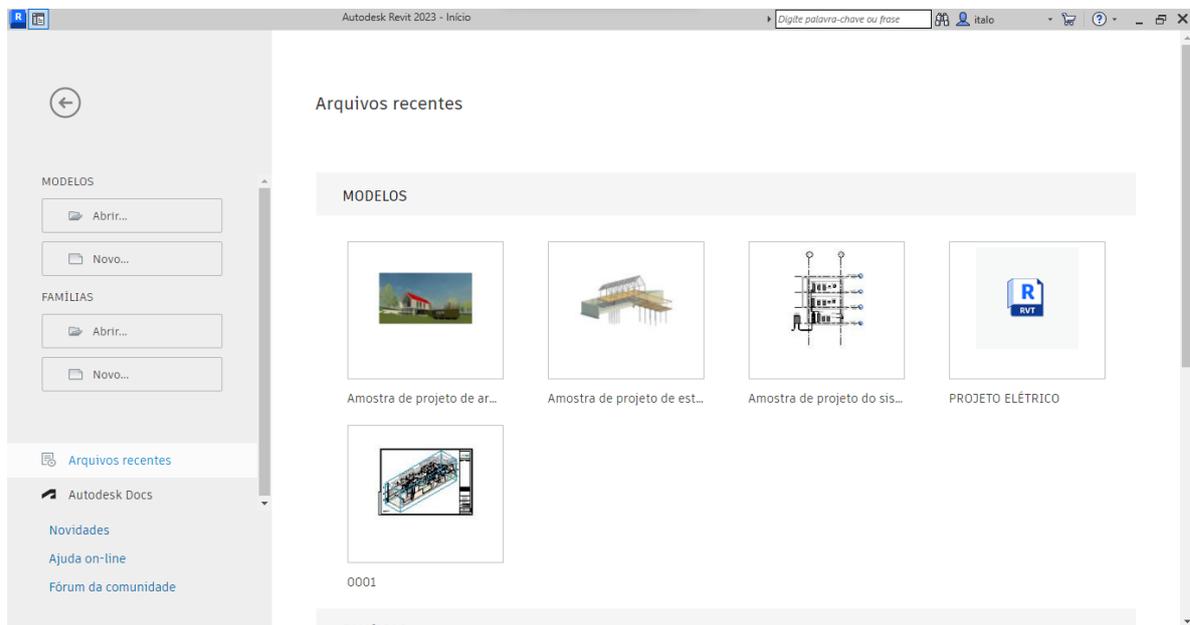
Caso o projetista já possua os modelos que atendem aos requisitos de projeto, a metodologia de construção dessas famílias tratada no presente trabalho pode ser desconsiderada e o profissional iniciaria suas atividades com a montagem dos elementos no projeto e o uso das automações, o que lhe economizaria horas ou dias com a modelagem adequada dos elementos, a depender do nível de detalhamento pretendido. Essa alternativa é o cenário ideal de atuação com projetos em BIM, que é mais rápido direto.

## **2.5 Visão geral e principais ferramentas do Revit**

A interface inicial do Revit 2023 pode ser vista na Figura 5, onde, na janela central, ficam os últimos arquivos abertos e alguns modelos padrão disponibilizados pela Autodesk. A princípio, é possível notar, na janela lateral esquerda, que a ferramenta discrimina o trabalho em duas vertentes principais para tipos de arquivos, os de modelo e os de família. O primeiro é, de fato, o arquivo do projeto, pois ele que se torna o modelo único mencionado em sessões anteriores – sendo discriminado pela extensão .RVT. O segundo é o arquivo que comportará um modelo virtual com todas as suas geometrias, parâmetros e autonomias – caracterizado pela extensão .RFA. Desse modo, uma família no Revit é um dos elementos que compõem um

modelo, ao passo que, um conjunto de padrões e características pré-definidas nos modelos qualificam-no como um *template* – de extensão .RTE, isto é, como um arquivo exemplo que atende à requisitos recorrentemente demandados e, por isso, são utilizados como base para projetos futuros, facilitando o trabalho do projetista.

Figura 5 – Interface inicial do Revit 2023.



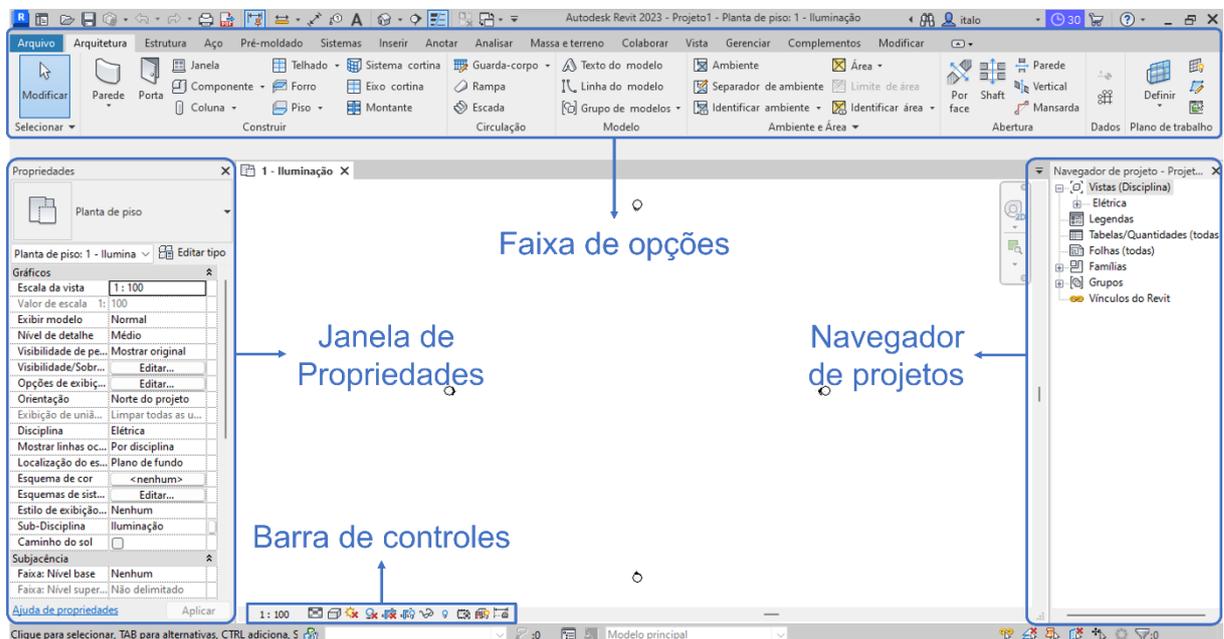
Fonte: o autor.

Quando se inicia um novo projeto, o *software* solicita que o usuário determine qual o modelo que se deseja seguir, dispondo de alguns padrões com perspectivas diferentes, como arquitetônica, mecânica, elétrica, dentre outras. Com essa ação, o usuário está definindo o *template* que comportará o projeto. Essa escolha não inibe o uso de atributos hidráulicos em um modelo elétrico, por exemplo, mas garante melhor visualização dos eletrodutos e dispositivos de proteção; tabelas com dimensionamentos automáticos para cabos; filtros específicos para elétrica nas vistas, dentre outras funcionalidades.

Criando-se um projeto, usando o *template* padrão para elétrica, um novo ambiente de trabalho é iniciado, do qual a interface pode ser vista na Figura 6, onde os principais itens são:

- Faixa de opções: separa em abas todas as ferramentas necessárias para criar e editar o projeto agrupadas por contexto;
- Janela propriedades: exibe os atributos (parâmetros) inerentes ao elemento selecionado, possibilitando editá-los quando permitido;
- Barra de controles da vista: dispõe de botões para definir o modo de exibição gráfica na tela e de escala para exportação;
- Navegador de projeto: hierarquiza todas as vistas segundo seu tipo (cortes, níveis, 3D, tabelas, legendas, folhas, famílias etc.). Várias vistas podem ficar abertas ao mesmo tempo e qualquer modificação em uma delas é efetuada simultaneamente em todas as outras.

Figura 6 – Interface de trabalho padrão no Revit 2023 para projetos elétricos.



Fonte: o autor.

### 2.5.1 Famílias no Revit

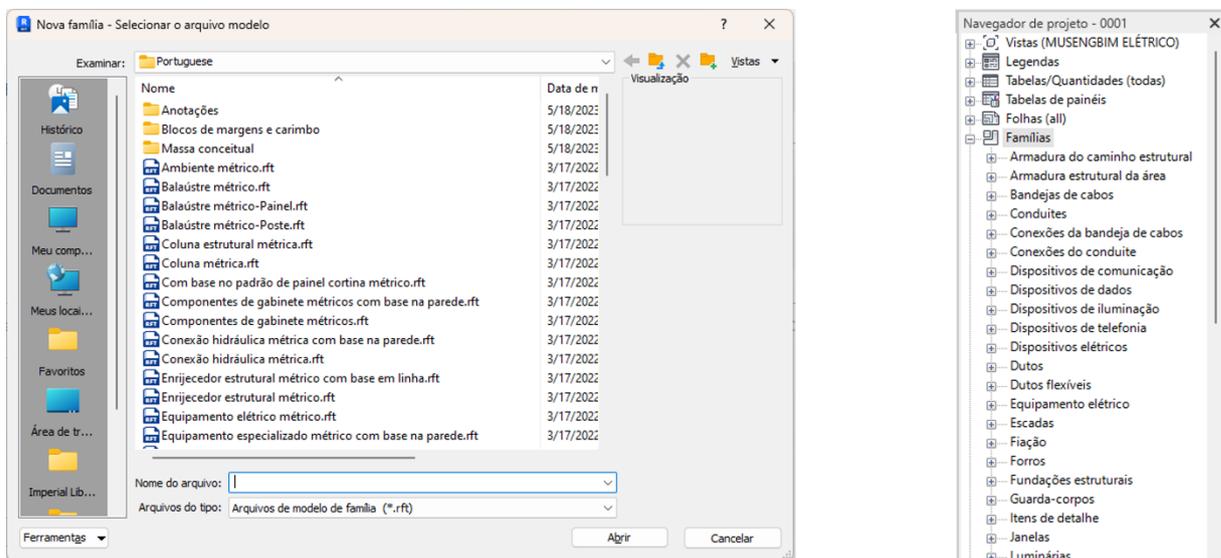
As famílias funcionam como entidades independentes, porém manipuláveis, incorporadas ao arquivo do projeto para compor as partes do empreendimento. Como exemplo, uma família de inversor certamente seria utilizada em um projeto solar fotovoltaico. Contudo, também são famílias os carimbos das folhas, os textos de

legenda, os elementos de detalhe, dentre outros. Nesse contexto, as famílias são divididas em três tipos:

- a. Família do sistema: são incorporadas nativamente ao *template*, não permitindo total modificação em suas características e não sendo acessíveis externamente por um arquivo .RFA. Um exemplo são os conduítes para cabos de energia;
- b. Família de anotação: são elementos bidimensionais (2D) criados para funcionarem como legendas ou *tags*;
- c. Família de modelo: são elementos criados externamente através de *templates* de família, podendo ser livremente editados e ter seus parâmetros modificados.

A categoria sob a qual cada família é criada define em qual grupo de elementos ela se encaixa e quais características ela deve acomodar. Atributos como esse permitem que o Revit seja eficaz na discriminação do conteúdo para filtragens e dimensionamentos em tempo real. Na Figura 7, à esquerda, pode-se ver algumas das categorias disponíveis. No lado direito é mostrado no navegador de projeto como encontrar as famílias já inseridas no arquivo principal. Nota-se que, além de classificar o elemento, em alguns casos, também se define o hospedeiro, isto é, a que classe de objeto a família será associada quando inserida no projeto. Como exemplo, inversores podem ser criados sob a categoria de equipamento elétrico com base na parede.

Figura 7 – Tipos de categorias para famílias criadas no Revit 2023.



Fonte: o autor.

### 2.5.1.1 Parâmetros e suas propriedades

Os parâmetros nada mais são que propriedades adicionadas aos elementos para manipular informações e definir características. Dentro do projeto, ao selecionar uma família, seus respectivos parâmetros podem ser acessados pela janela de propriedades, mostrada anteriormente. No Revit, existem duas classes de parâmetros, os compartilhados e os da família. O primeiro é acessado em tabelas, utilizado para transpor informações entre diversas famílias ou entre famílias e o projeto. Já o segundo se trata de parâmetros que carregam características inerentes à cada família e que devem ser classificados em duas categorias:

- a. Parâmetro de tipo: é o atributo ao qual seu valor característico está associado ao tipo. Isso implica que todas as famílias de determinado tipo inseridas no projeto contenham a mesma informação. Esses parâmetros são acessados através do campo “Editar tipo” na janela de propriedades e, ao alterar o valor de algum, o mesmo valor é modificado em todas as outras famílias do mesmo tipo. Como exemplo, ao corrigir o valor da corrente de curto-circuito em apenas um módulo de um projeto com 242 desse equipamento, todos os outros são ajustados automaticamente;
- b. Parâmetro de instância: é o atributo ao qual seu valor característico está associado a cada elemento (instância) individualmente, sem influenciar nos outros. Esses parâmetros são acessados diretamente na aba principal da janela de propriedades. Como exemplo, considerando os mesmos 242 módulos anteriores, cada módulo deve conter uma identificação única, referente ao inversor que está conectado, à MPPT, à *string* e qual sua posição dentro dela.

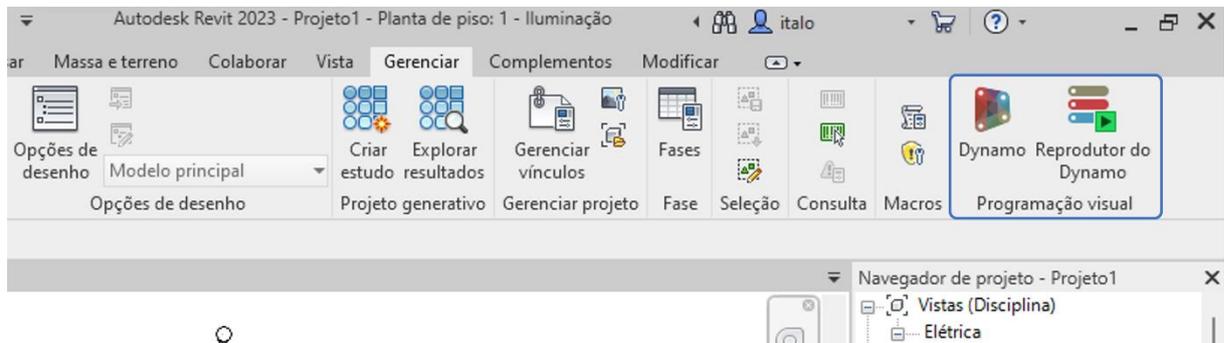
## 2.6 Interfaces de programação no Dynamo

### 2.6.1 Visão geral do ambiente

A Autodesk disponibiliza a instalação do Dynamo conjuntamente com a do Revit, permitindo que o primeiro seja acessado diretamente pelo aplicativo principal. Para isso, com qualquer projeto aberto, basta abrir a aba “Gerenciar”, na parte superior, e

no painel “Programação visual” encontra-se o *software*, como indicado na Figura 8. O mesmo painel contém também a ferramenta “Reprodutor do Dynamo”, que se trata de uma interface de usuário, para que os projetistas possam visualizar e usar, de forma rápida e simples, todas as automações, sem necessariamente acessar o *script*.

Figura 8 – Acessando o Dynamo e sua interface de usuário via Revit 2023.

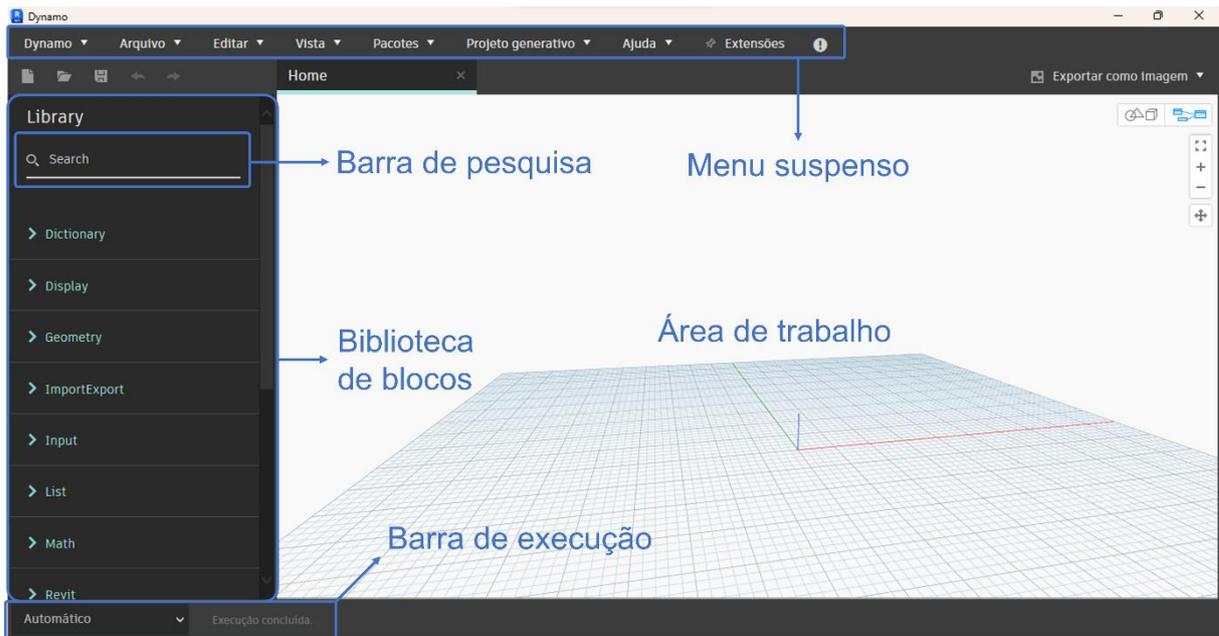


Fonte: o autor.

A interface principal do Dynamo pode ser vista na Figura 9, onde os itens em destaque são:

- a) Menu suspenso: fornece algumas funcionalidades básicas como a gestão de arquivos, ajustes de configurações, controle de pacotes, dentre outras;
- b) Barra de pesquisa: permite a busca por blocos disponíveis para programação visual com *DesignScript*;
- c) Biblioteca de blocos: essa biblioteca separa as subcategorias dos blocos – também chamados de *nodes*. Através da importação de pacotes externos é possível inserir novos *nodes* com funcionalidades diversas;
- d) Barra de execução: onde se encontra o botão de execução do algoritmo. Nessa região também fica visível a situação do *script* após sua atuação;
- e) Área de trabalho: é a região onde é construído o algoritmo, utilizando linguagem visual, textual ou ambas. Além disso, esse espaço pode ser alternado para a vista gráfica 3D dos resultados, que fica no plano de fundo e pode ser acessada através dos botões de navegação localizados no canto superior direito.

Figura 9 – Interface principal do Dynamo.



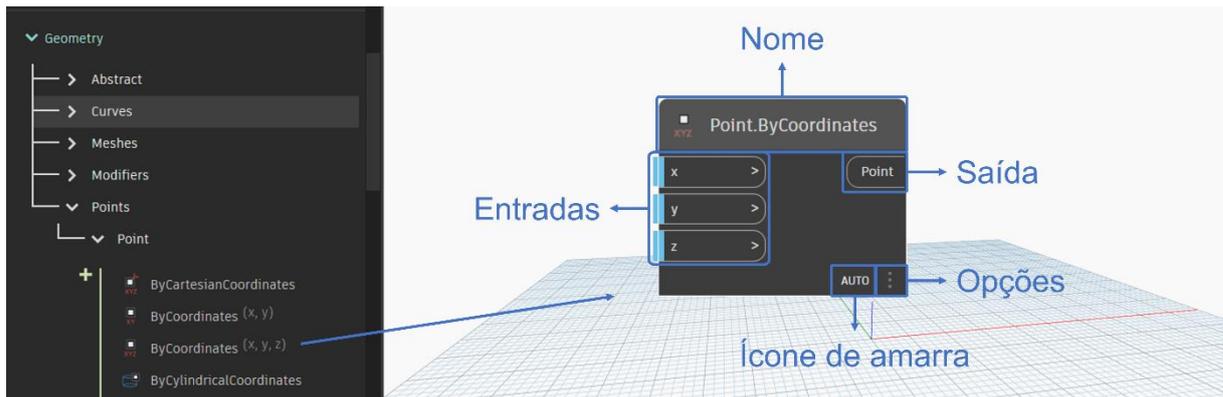
Fonte: o autor.

### 2.6.2 Definições e operação com o Dynamo

Para operar no Dynamo é importante entender a arquitetura dos *nodes* e como eles operam relacionando-se entre si. A Figura 10 indica a “anatomia” de um bloco genérico da biblioteca *Geometry* – usada para trabalhar com geometrias, que cria um ponto no espaço cartesiano do Revit seguindo as coordenadas fornecidas. De modo geral, as entradas de um *node* funcionam como *inputs* para sua função interior e, após processadas, os dados resultantes são transmitidos pela saída, podendo ser visualizados na parte inferior do *node*.

Blocos no Dynamo podem conter várias portas de entrada, mas apenas uma de saída. Sendo assim, o desenvolvedor manipula o formato de saída desejado para os dados e essa definição fica visível no ícone de amarra dos resultados. No botão de opções é possível definir todas as características do *node*, tanto gráficas – como a visibilidade dele e o agrupamento com outros *nodes*, quanto funcionais – como inativar o bloco ou definir se ele deve colher ou disponibilizar dados na interface de usuário.

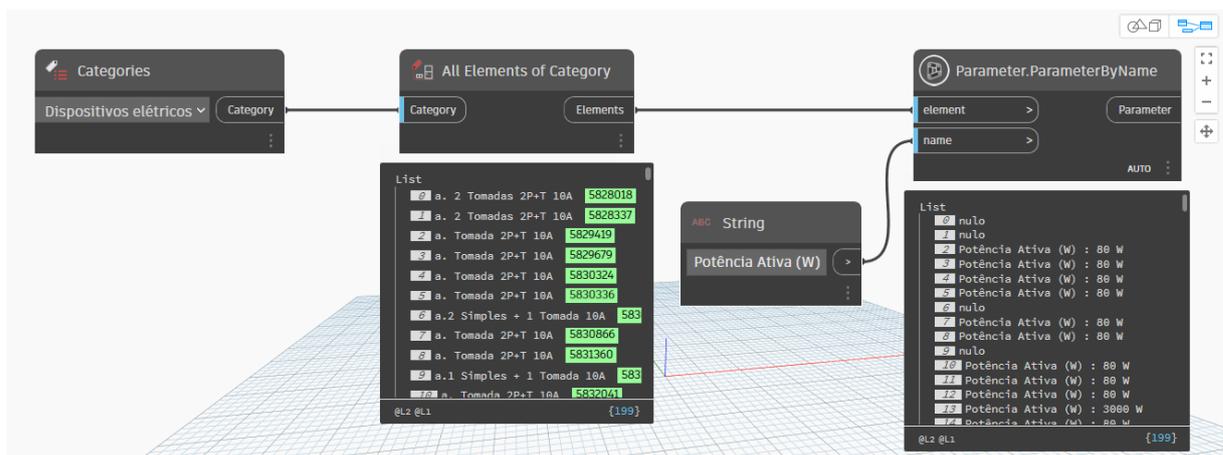
Figura 10 – Exemplo genérico de bloco do Dynamo.



Fonte: o autor.

Acessando um projeto elétrico de baixa tensão (BT) comum, a sequência apresentada na Figura 11 é formada por blocos básicos do Dynamo. Nesse curto algoritmo é possível rastrear todos os dispositivos elétricos no empreendimento, com o *node* “*All elements of category*”, e consultar a potência ativa de cada um deles. A conexão entre *nodes* é feita diretamente clicando entre entradas e saídas, em qualquer ordem. Naturalmente, como as entradas estão localizadas no lado esquerdo e as saídas no direito, a sequência de execução dos códigos sempre será da esquerda para a direita, de modo que os dados de saída de um *node* sejam de entrada em um ou mais subsequentes. Nos 199 elementos encontrados no exemplo apresentado existem tipos repetidos, o que é esperado, mas nunca instâncias com o mesmo número identificador (ID), destacado em verde, pois esse número é único.

Figura 11 – Algoritmo básico no Dynamo para consulta da parâmetros elétricos.



Fonte: o autor.

### 2.6.3 Python e API do Revit

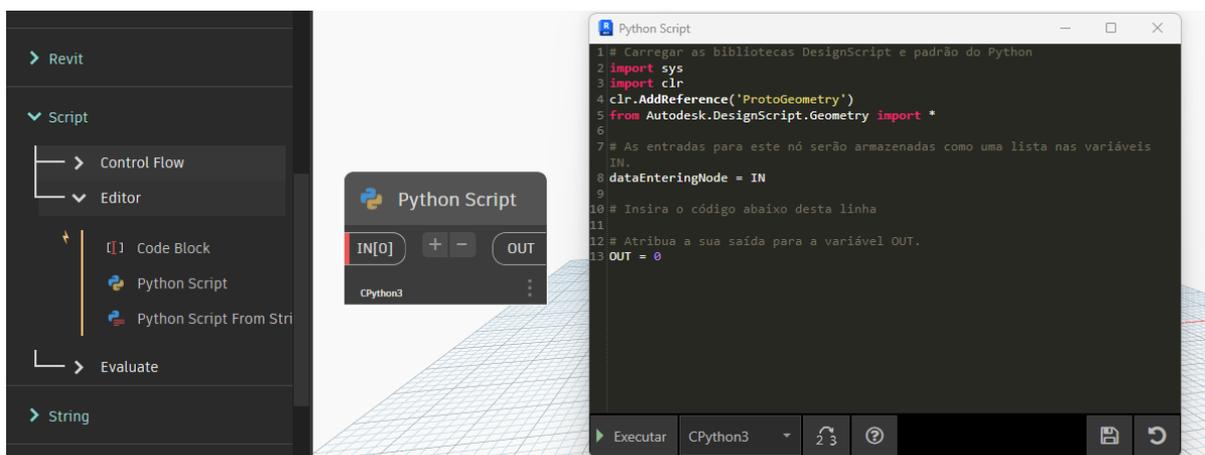
Como já comentado anteriormente, o modo de programação textual também é permitido no Dynamo e ele proporciona mais liberdade ao desenvolvedor, pois todas as funções tornam-se acessíveis e manipuláveis. Usando essa alternativa, é possível realizar através de poucas linhas de código o equivalente a uma complexa rede de *nodes* conectados, tornando, dessa forma, o programa limpo.

Para usar o Python e a API do Revit é preciso usar um bloco de *script*, a esquerda na Figura 12. Não faz parte do escopo do presente trabalho discorrer especificamente sobre as atividades realizadas pelas funções utilizadas nos *scripts* das rotinas desenvolvidas.

Figura 12, através do qual é possível acessar a janela de programação, a direita. Nesse caso, então, o programador é quem define a quantidade de entradas do bloco através dos botões centrais. No código, as entradas são acessadas com a variável *built-in* “IN”, enquanto a saída é definida com “OUT”.

É permitido importar facilmente algumas bibliotecas externas em Python, assim como todas as funções da API, visto que a Autodesk disponibiliza todas as versões das documentações de forma online em site aberto [17]. Não faz parte do escopo do presente trabalho discorrer especificamente sobre as atividades realizadas pelas funções utilizadas nos *scripts* das rotinas desenvolvidas.

Figura 12 – Modelo padrão para bloco de programação em Python no Dynamo.



Fonte: o autor.

## **3 FAMÍLIAS REVIT E AUTOMAÇÕES NO DYNAMO**

### **3.1 Objetos para desenvolvimento**

Mesmo com a escassez de produtos BIM disponíveis para serem utilizados em projetos solares fotovoltaicos, um dos pilares a serem seguidos no presente trabalho é a realização da modelagem de famílias inteligentes e funcionais no Revit, seguindo as diretrizes apresentadas anteriormente para parametrização e compatibilização com aplicações externas. O subsequente pilar vem a ser o desenvolvimento de duas rotinas, que são, para o dimensionamento de dispositivos, para a montagem de quadros elétricos, para a alocação de circuitos e para a construção de diagramas trifilares, tudo em concordância com as famílias previamente construídas.

### **3.2 Construção das famílias parametrizadas**

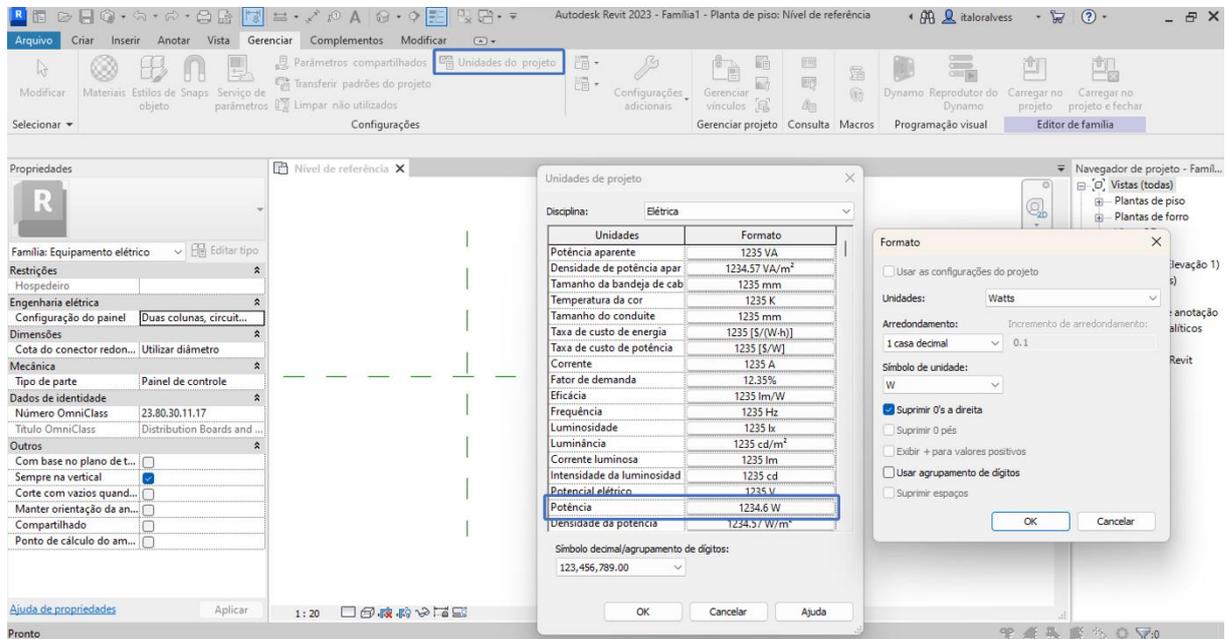
#### **3.2.1 Família de inversor**

Para atender as automações, serão demonstrados os métodos para a modelagem de famílias paramétricas 3D e 2D no Revit, abordando de forma global e estratégica os procedimentos. Para o primeiro caso, o objeto de estudo serão inversores da Sungrow, uma das fabricantes que mais fornece equipamentos para usinas solares fotovoltaicas (UFVs) instalados no Brasil. Os modelos escolhidos foram os da linha CX com potências de 40 kW e 50 kW, que são construídos sob a mesma estrutura. Suas especificações técnicas, são retiradas do manual do usuário e do *datasheet*.

Seguindo o processo descrito nas subseções anteriores, a família de inversores deve ser criada sob a categoria de equipamentos elétricos. A primeira coisa a se fazer ao iniciar um novo projeto é definir suas unidades de medida padrão, para que estejam de acordo com o objeto real, facilitando também a construção do modelo e a possível representação de cotas. Através da ferramenta “Unidades do projeto”, no painel “Configurações” da aba “Gerenciar”, é possível controlar todas as unidades que o

Revit comporta. A Figura 13 demonstra o acesso para o caso das unidades de potência.

Figura 13 – Controle de unidades do projeto no Revit.



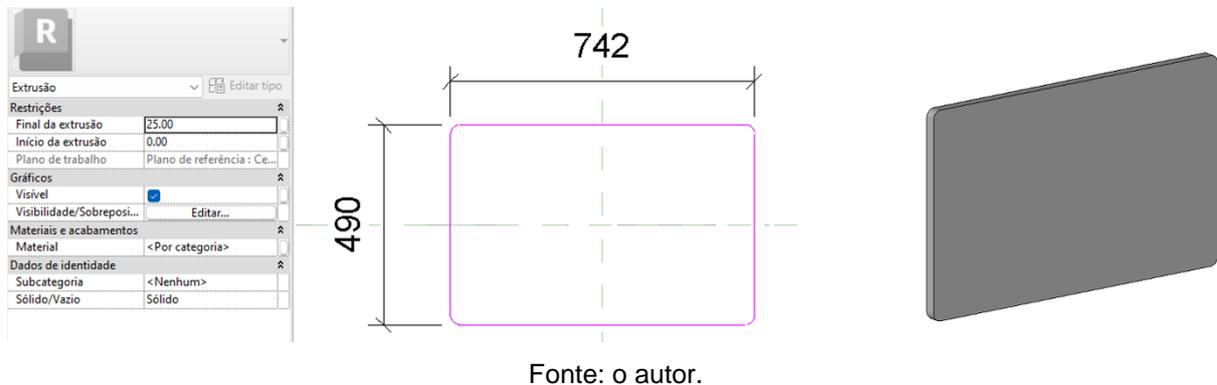
Fonte: o autor.

A área de trabalho inicial é uma vista do nível de referência, ou seja, é uma perspectiva superior 2D, semelhante à uma planta baixa. Essa vista conta com duas linhas verdes tracejadas e perpendiculares, que são a representação bidimensional de planos de referência espaciais que dividem o modelo em partes frontal/traseira e esquerda/direita. Esses elementos são fundamentais para a orientação e controle gráfico do modelo. Para visualizar a construção no decorrer da modelagem, será preciso transitar constantemente pelas diferentes vistas no navegador de projeto.

A modelagem inicia-se com troca para a elevação da parte frontal, seguida da representação detalhada do equipamento, em partes, no Revit. Optando-se por iniciar representando a tampa frontal do inversor, é preciso usar o comando “Extrusão” no painel “Formas” da aba “Criar”, para gerar o sólido seguindo a conformação e as dimensões do equipamento, como indicado no lado esquerdo da Figura 14. Percebe-se que o diagrama respeita as dimensões de largura e altura, enquanto a profundidade

da extrusão é definida pelo parâmetro “Final da extrusão”, na janela de propriedades. No lado direito da mesma imagem encontra-se o resultado tridimensional.

Figura 14 – Criando sólidos para construção de famílias.



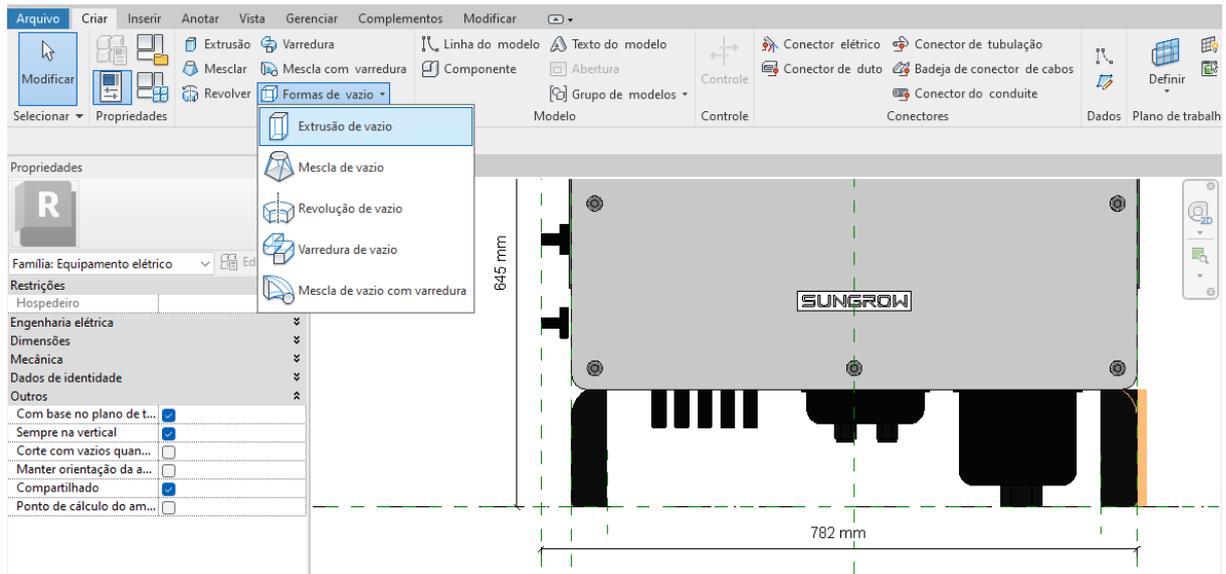
Fonte: o autor.

Aspectos gráficos de preenchimento no Revit não são apenas cores e texturas, mas sim acabamentos com definições específicas do material. Dessa forma, para a representação do inversor ficar completa, é preciso alimentar a biblioteca com os diferentes materiais que o compõe, através da ferramenta “Materiais”, presente também na Figura 13. A criação desses materiais foge do escopo do trabalho.

Além da criação de várias formas por extrusão, por varredura, por revolução e todas as outras possibilidades de simetrias disponíveis, a composição de geometrias mais complexas pode ser feita através de ajustes, “cortando” esses sólidos com formas em vazio, como exemplificado na Figura 15, onde a região de corte, que foi criada como uma extrusão em vazio, está indicada em laranja, esculpindo as curvaturas das alças inferiores do equipamento.

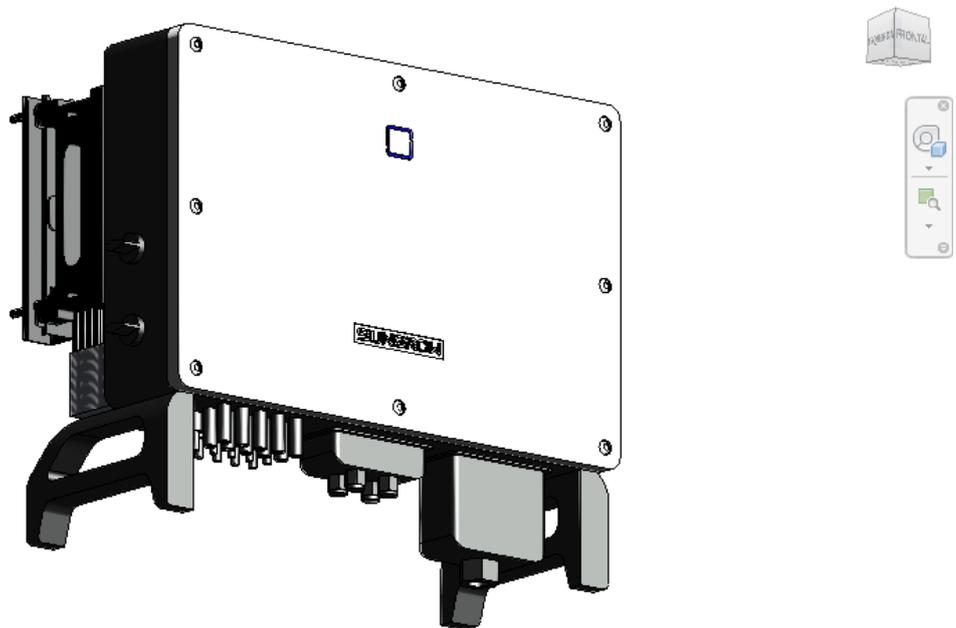
Quanto mais elaborado for o modelo pretendido mais geometrias em vazio precisarão ser usadas para moldar as famílias. A Figura 16 mostra o resultado gráfico final para a família dos inversores Sungrow que atende aos modelos SG40CX e SG50CX, com todos os seus detalhamentos construtivos e respeitando as dimensões reais do elemento.

Figura 15 – Extrusão vazia dando forma a geometria complexa.



Fonte: o autor.

Figura 16 – Família Revit de inversor Sungrow para modelos SG40CX e SG50CX.



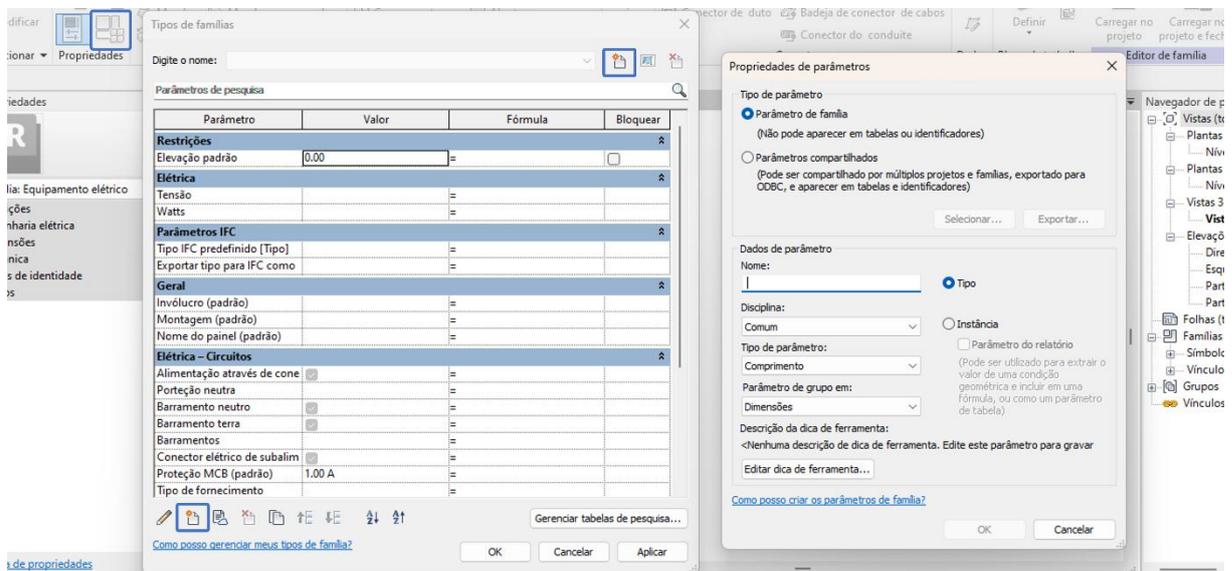
Fonte: o autor.

A segunda parte, e a principal delas, é a parametrização da família. Para criar, editar ou excluir parâmetros de uma família do Revit basta acessar a ferramenta “Tipos de famílias”, presente no painel “Propriedades” da aba “Criar”, como indicado na Figura 17. Por padrão, alguns parâmetros são previamente criados, porém sem valores atribuídos. É possível criar um parâmetro através do botão indicado no canto

inferior esquerdo da Figura 17. O fluxo dentro da janela de “Propriedades de parâmetros” inicia com a definição quanto ao tipo, se o parâmetro será de família ou compartilhado, sendo o segundo caso não abordado no presente trabalho. Dessa forma, após classificar o parâmetro a ser criado como de família, na área “Dados de parâmetro” é possível determinar suas características de acordo com os seguintes critérios:

- Nome: definir uma nomenclatura única para o parâmetro;
- Disciplina: classifica a área de enquadramento do parâmetro e, dessa forma, define a disponibilidade de tipos no tópico seguinte. Exemplo: elétrica, estrutural, energia, infraestrutura etc.;
- Tipo de parâmetro: atribui significado ao parâmetro, isto é, determina a medida ou composição representada. Exemplo: número inteiro, texto, velocidade, tensão elétrica, luminosidade etc.;
- Parâmetro de grupo em: tem o objetivo de organizar o quadro de parâmetros na janela de acesso do usuário;
- Tipo ou Instância: determina a categoria do parâmetro.

Figura 17 – Criação de parâmetros de família no Revit.



Fonte: o autor.

Um ponto importante para atenção é quanto a definição da disciplina e do tipo de parâmetro, pois são campos que não podem ser alterados posteriormente. Para que ambos os modelos do inversor Sungrow sejam atribuídos à essa mesma família eles devem ser distinguidos por tipos diferentes. Para criar um tipo basta selecionar o ícone indicado no canto superior direito da janela “Tipos de famílias” na Figura 17 e nomeá-lo.

Quando uma família está em utilização dentro de um projeto, qualquer parâmetro de tipo pode armazenar valores diferentes para tipos distintos, ao passo que o mesmo não acontece para parâmetros de instância em tipos diferentes de uma mesma instância. Todos os parâmetros criados para a família de inversores estão descritos na Tabela 2. No tópico destinado para análise dos resultados serão apresentadas as outras famílias paramétricas criadas para o presente trabalho, sendo elas a de módulo, de disjuntor, de DPS e dos diagramas trifilares.

Tabela 2 – Descrição de parâmetros da família de inversores Sungrow modelos SG40CX e SG50CX.

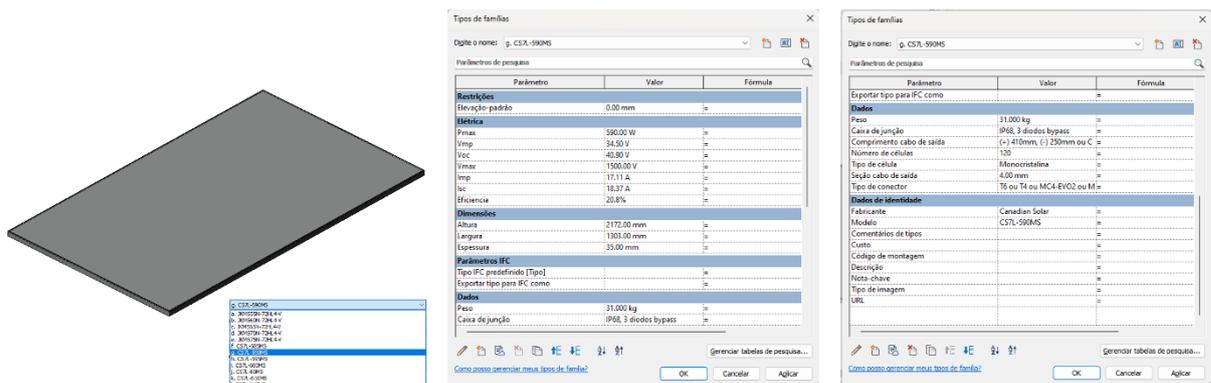
| Nome          | Disciplina | Tipo de parâmetro  | Grupo               | Categ. | SG40CX  | SG50CX  |
|---------------|------------|--------------------|---------------------|--------|---------|---------|
| Fabricante    | Comum      | Texto              | Dados de identidade | Tipo   | Sungrow | Sungrow |
| Modelo        | Comum      | Texto              | Dados de identidade | Tipo   | SG40CX  | SG50CX  |
| Chave_CA      | Comum      | Sim/Não            | Construção          | Tipo   | Não     | Não     |
| Chave_CC      | Comum      | Sim/Não            | Construção          | Tipo   | Sim     | Sim     |
| DPS_CA        | Comum      | Sim/Não            | Construção          | Tipo   | Sim     | Sim     |
| DPS_CC        | Comum      | Sim/Não            | Construção          | Tipo   | Sim     | Sim     |
| Strings/MPPT  | Comum      | Número inteiro     | Construção          | Tipo   | 2       | 2       |
| Total_MPPTs   | Comum      | Número inteiro     | Construção          | Tipo   | 4       | 5       |
| Classe_DPS_CA | Comum      | Texto              | Elétrica            | Tipo   | II      | II      |
| Classe_DPS_CC | Comum      | Texto              | Elétrica            | Tipo   | II      | II      |
| Corr_max_CA   | Elétrica   | Corrente           | Elétrica            | Tipo   | 66.9    | 83.6    |
| Corr_max_CC   | Elétrica   | Corrente           | Elétrica            | Tipo   | 104     | 130     |
| Eficiencia    | Elétrica   | Texto              | Elétrica            | Tipo   | 98.6    | 98.7    |
| Fases         | Elétrica   | Número de polos    | Elétrica            | Tipo   | 3       | 3       |
| Frequencia    | Elétrica   | Frequência         | Elétrica            | Tipo   | 60      | 60      |
| Potencia      | Elétrica   | Potência           | Elétrica            | Tipo   | 40000   | 50000   |
| THD           | Comum      | Texto              | Elétrica            | Tipo   | <3%     | <3%     |
| Tens_CA_inter | Comum      | Texto              | Elétrica            | Tipo   | 312-528 | 312-528 |
| Tens_CA_nom   | Elétrica   | Potencial elétrico | Elétrica            | Tipo   | 380     | 380     |

| Nome           | Disciplina | Tipo de parâmetro  | Grupo    | Categ. | SG40CX          | SG50CX          |
|----------------|------------|--------------------|----------|--------|-----------------|-----------------|
| Tens_FV_max    | Elétrica   | Potencial elétrico | Elétrica | Tipo   | 1100            | 1100            |
| Tens_FV_nom    | Elétrica   | Potencial elétrico | Elétrica | Tipo   | 585             | 585             |
| Tens_MPP_inter | Comum      | Texto              | Elétrica | Tipo   | 550-800         | 550-800         |
| Comunicacao    | Comum      | Texto              | Geral    | Tipo   | RS485/<br>Wi-Fi | RS485/<br>Wi-Fi |
| Grau_protecao  | Comum      | Texto              | Geral    | Tipo   | IP66            | IP66            |
| Temperat_max   | AVAC       | Temperatura        | Geral    | Tipo   | 60              | 60              |
| Temperat_min   | AVAC       | Temperatura        | Geral    | Tipo   | -30             | -30             |

Fonte: o autor.

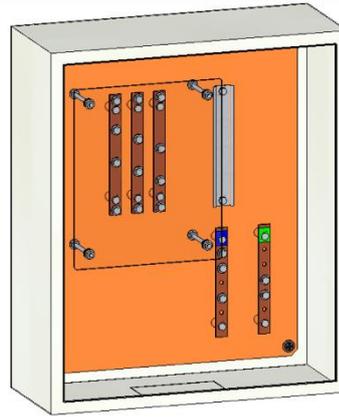
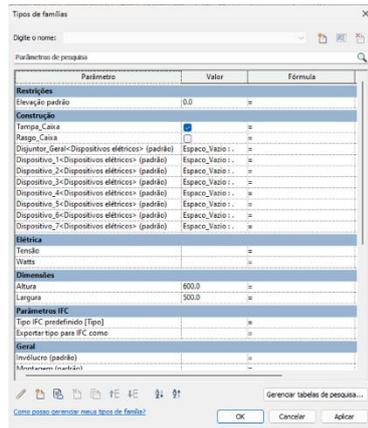
Outro tipo de família desenvolvido foi o de módulo fotovoltaico (FV), que conta com uma série de parâmetros elétricos preenchidos com os respectivos valores de seu *datasheet*. Para o caso de estudo, as fabricantes selecionadas foram a Jinko Solar e a Canadian Solar, com equipamentos de potências nominais de 555 Wp até 615 Wp, como pode ser visto na Figura 18. Outros modelos desenvolvidos para o caso de estudo (além do inversor, que já foi detalhado Seção 3.2.1), bem como seus parâmetros, estão ilustradas na Figura 19 para o caso de modelos tridimensionais, atendendo às categorias de equipamentos elétricos, dispositivos elétricos e modelos genéricos. É importante relatar que, além dessas, outras famílias não destacadas precisaram ser criadas para o funcionamento correto dos modelos e estão aninhadas às principais.

Figura 18 – Família paramétrica de módulos com tipos para diversos modelos.

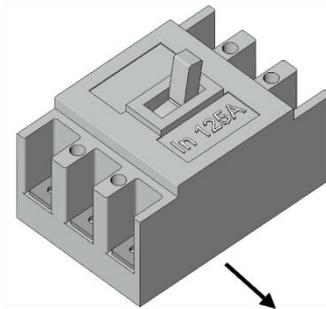
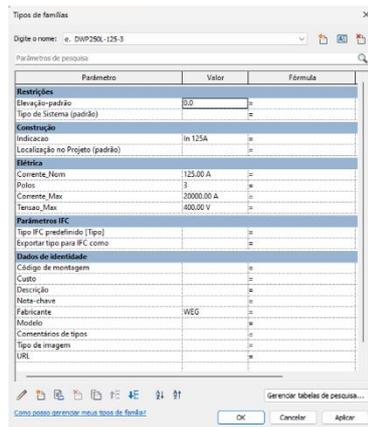


Fonte: o autor.

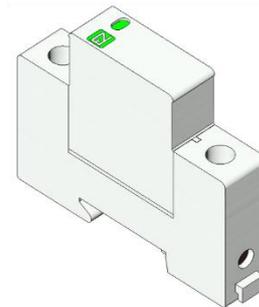
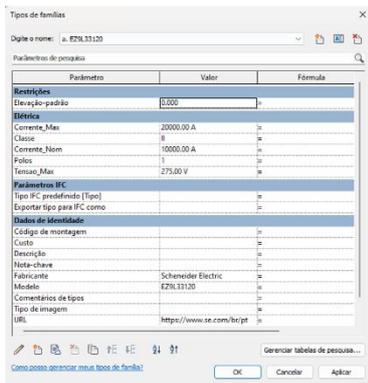
Figura 19 – Principais famílias criadas para projetos e automações.



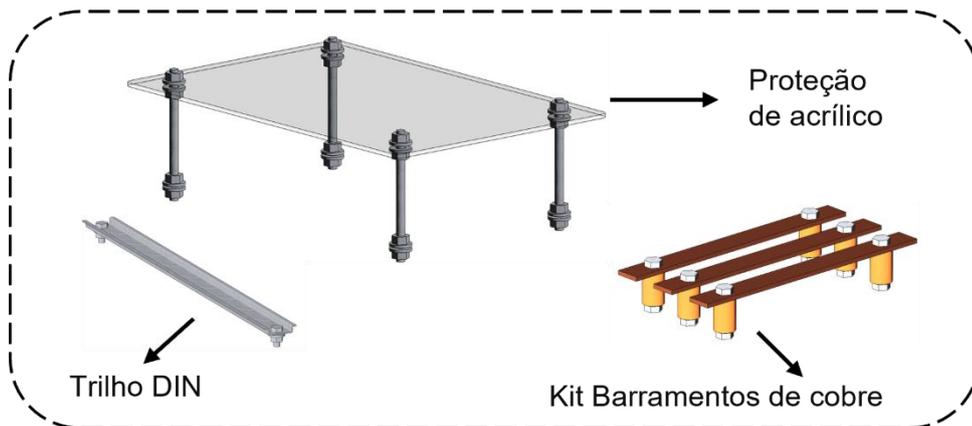
Quadro metálico



Disjuntor Caixa Moldada



DPS CA



Trilho DIN

Kit Barramentos de cobre

Proteção de acrílico

Fonte: o autor.

### 3.2.2 Família de diagrama trifilar do inversor

Para a criação das famílias de diagramas trifilares é usado como base o template de modelo genérico métrico com base em dois níveis, pois permite a utilização de materiais de diferentes categorias em sua composição e a família final pode ser visualizada em qualquer vista do projeto. Nesta etapa, apesar da modelagem da família ser mais simples, por se tratar de representações 2D com linhas, a parametrização é mais elaborada. Essa premissa é necessária ao passo que os diagramas precisam se adequar ao modelo virtual 3D, isto é, devem se ajustar para representar graficamente um inversor com 2 *strings* em cada uma das 5 MPPTs para o SG50CX do caso anterior, por exemplo. Essa liberdade é possível com a aplicação de algumas técnicas, como o controle de visibilidade – que determina quais elementos estarão visíveis ou ocultos, e o uso de elementos móveis controlados por cotas, como planos de referência. Essas ferramentas são conduzidas por operações aritméticas e condições inseridas na fórmula do parâmetro. Na Tabela 3 e na Tabela 4 estão indicadas as principais operações aritméticas e lógicas bem como as condições utilizadas nas famílias que serão posteriormente desenvolvidas

Tabela 3 – Operações aritméticas utilizadas nas fórmulas dos parâmetros das famílias.

| Operações aritméticas |         |
|-----------------------|---------|
| Descrição             | Fórmula |
| Adição                | +       |
| Subtração             | -       |
| Multiplicação         | *       |
| Divisão               | /       |
| Valor absoluto        | abs( )  |
| Maior que             | >       |
| Menor que             | <       |

Fonte: o autor.

Tabela 4 – Condições e operações lógicas utilizadas em fórmulas dos parâmetros das famílias.

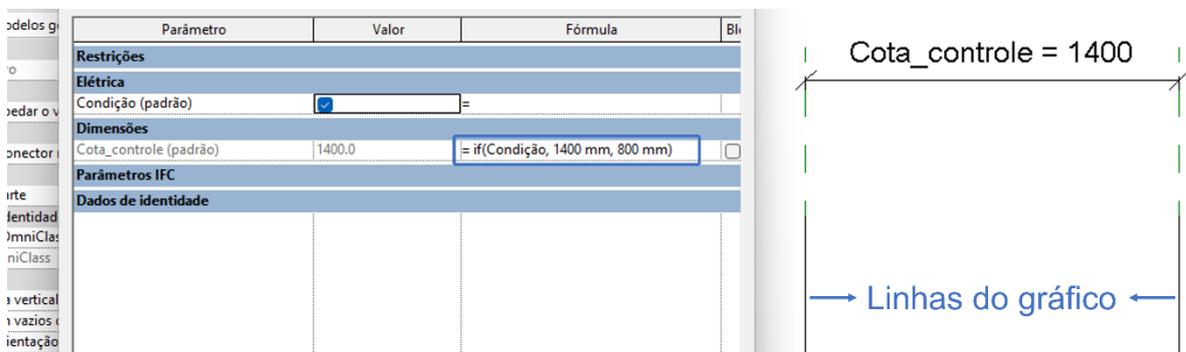
| Condições e operações lógicas |                                      |
|-------------------------------|--------------------------------------|
| Descrição                     | Fórmula                              |
| Arquitetura da condicional E  | and(<condição 1>, <condição 2>, ...) |
| Arquitetura da condicional OU | or(<condição 1>, <condição 2>, ...)  |

| Condições e operações lógicas |   |
|-------------------------------|---|
| Descrição                     | Fórmula   |
| Arquitetura para negação      | not(<valor>)  |
| Arquitetura da condicional SE | if(<condição>, <resultado se verdadeiro>, <resultado se falso>) |
| SE com base em parâmetro      | if(selecione, “Com DPS”, “Sem DPS”)                             |
| SE com análise de parâmetro   | if(comprimento > 10 mm, 18 mm, 7 mm)                            |
| SE com condição integral      | if(and(i = 66.9 A, v = 380 V), “SG40CX”, “SG50CX”)              |
| SE com condição parcial       | if(or(potencia = 40000, pot = 50000), 60 Hz, 0 Hz)              |
| SE com condições em cascata   | if(cota < 50 mm, 30 mm, if(cota < 80 mm, 65 mm, 100 mm))        |

Fonte: o autor.

Como exemplo, na Figura 20 é demonstrada a condição que determina o espaçamento entre dois planos de referência de acordo com o parâmetro “Condição”, sendo de 1400 mm quando verdadeiro ou 800 mm quando falso. Esse parâmetro foi atribuído à sua respectiva cota através da ferramenta “Legenda”, presente no painel “Cota da Legenda” da aba “Modificar | Cotas”. Os planos de referência “carregam” as linhas de representação, ou seja, os planos são elementos condutores e as linhas elementos conduzidos, que são a parte visível externamente à família.

Figura 20 – Demonstração do uso de condições em parâmetros de família.

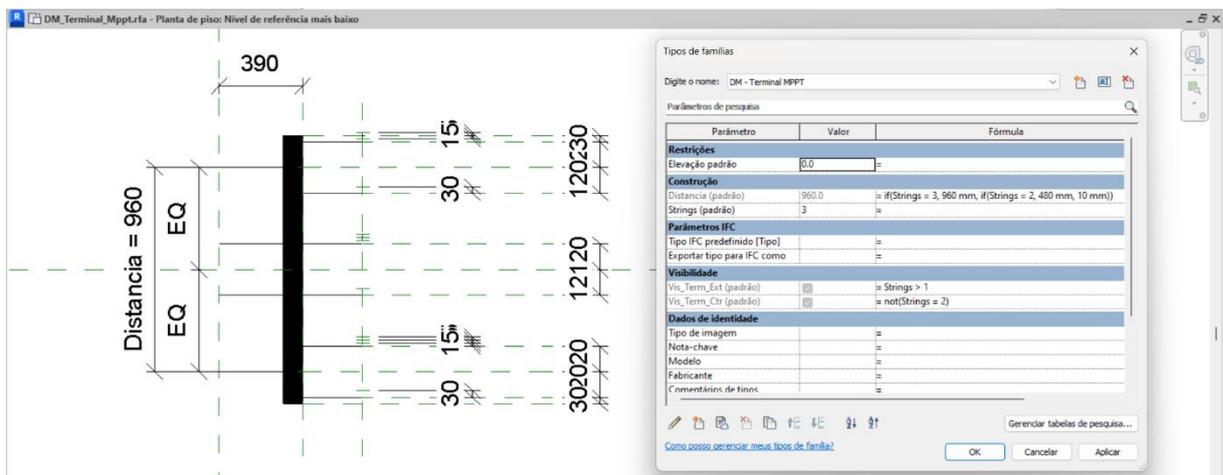


Fonte: o autor.

Para o correto funcionamento dos diagramas, e demais famílias complexas aqui desenvolvidas, optou-se por utilizar o atributo *Nested Families* do Revit, o qual implica

no carregamento de famílias específicas dentro de outras mais genéricas. Esse artifício confere à família hospedeira controle sobre os parâmetros do modelo secundário. Iniciando a construção do diagrama trifilar do inversor pela representação das MPPTs como uma família específica, foi escolhido abordar nesses instrumentos entradas para três *strings* em paralelo, abarcando assim outros possíveis modelos de inversores. O controle das linhas é feito através das cotas e parâmetros conforme Figura 21.

Figura 21 – Controle do diagrama trifilar de uma MPPT com três *strings*.



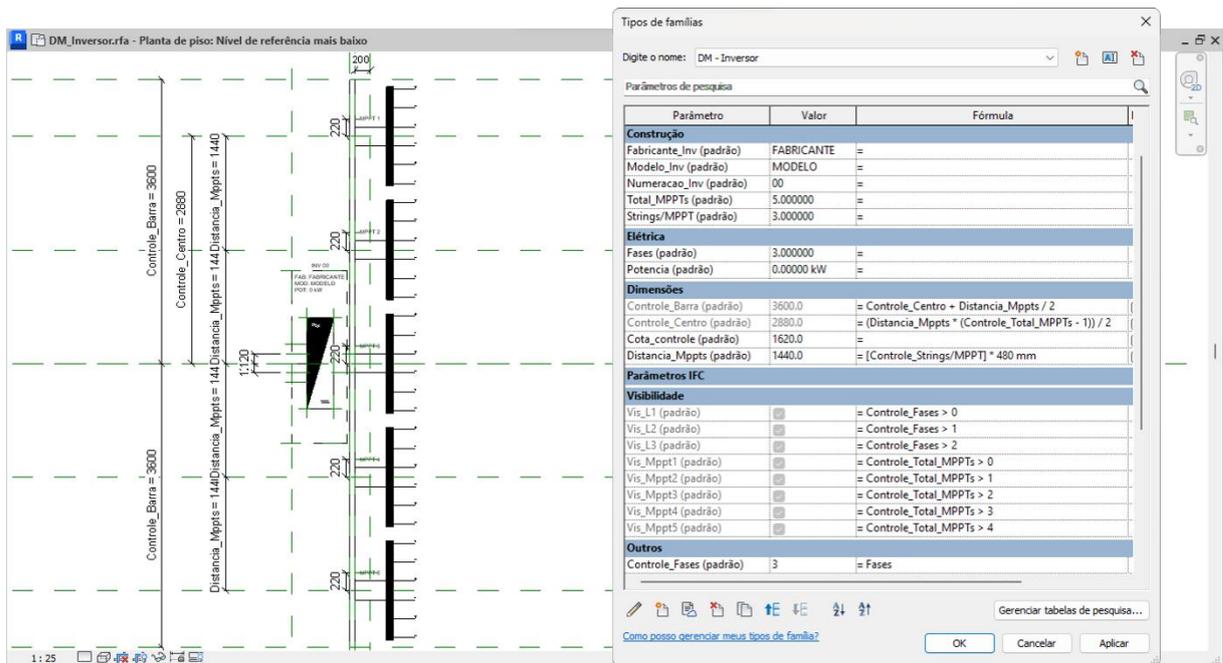
Fonte: o autor.

Em seguida, é criada a família principal, que hospedará essa primeira. Para tal, basta usar a ferramenta “Carregar no projeto”, presente no painel “Editor de família” da aba “Criar”, e selecionar a família hospedeira. Recomenda-se sempre selecionar a opção de carregar a família e todos os seus valores de parâmetro para garantir que todas as alterações foram atualizadas. Nesta, são construídas o restante das representações gráficas necessárias para um inversor, juntamente com cinco dos terminais MPPT, e os parâmetros, como mostrado na Figura 22.

A Figura 23 indica que ao lado dos parâmetros de instância das famílias secundárias há um botão retangular que permite sua associação com parâmetros da família principal que sejam do mesmo tipo, proporcionando assim, o controle do segundo sobre o primeiro. Acontece de forma análoga para atribuir o comando de visibilidade a um elemento qualquer, onde o parâmetro deve ser do tipo “Sim/Não” e

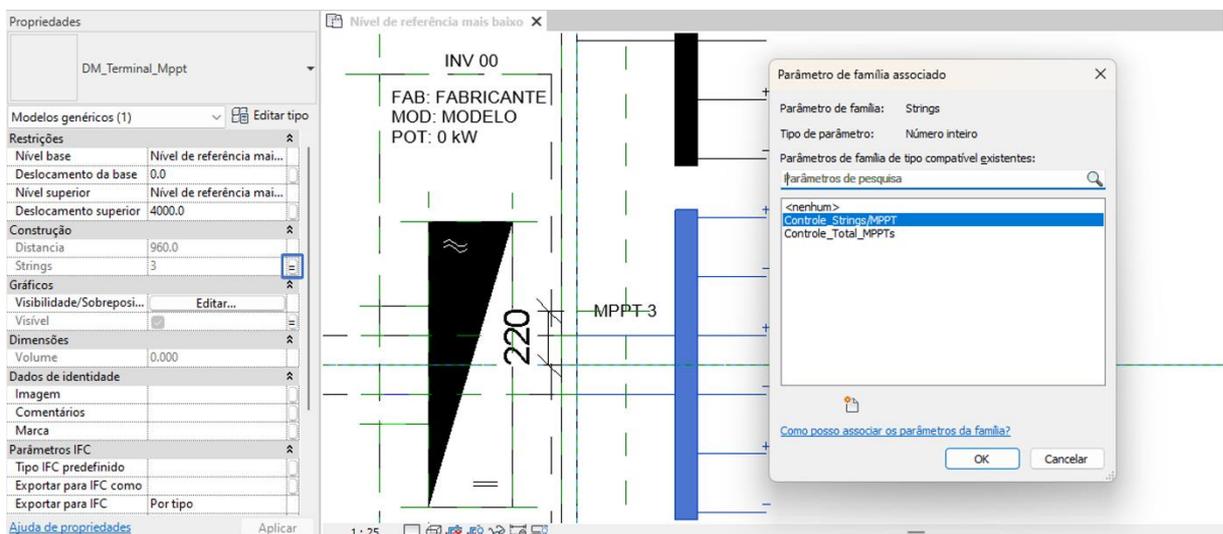
pode ser controlado por fórmula assim como os demais, com atenção para que o resultado da condição seja sempre um *booleano*. A legenda do inversor é formada por duas famílias de “Anotações genéricas”, onde cada parte do texto é alimentada por seu respectivo parâmetro, os quais têm seus valores comandados por atribuições semelhantes à demonstrada anteriormente.

Figura 22 – Construção e controle do diagrama trifilar de inversor fotovoltaico.



Fonte: o autor.

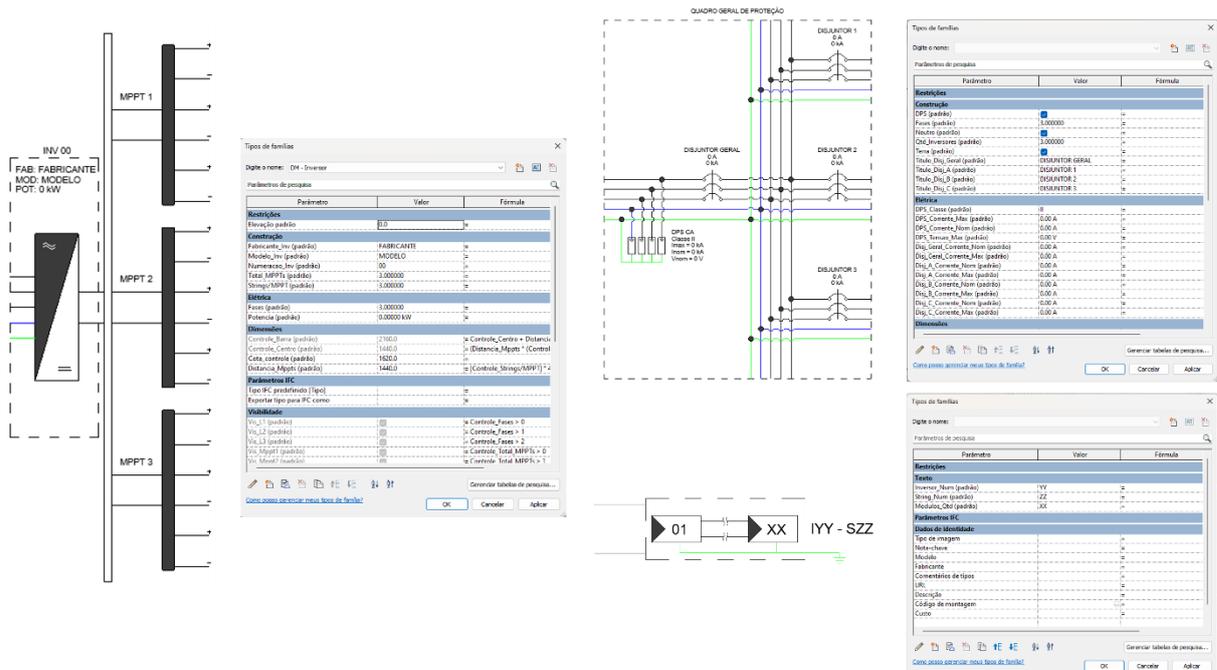
Figura 23 – Associação de parâmetros entre famílias aninhadas.



Fonte: o autor.

Seguindo essa metodologia, foram desenvolvidas famílias totalmente ajustáveis para diagramas trifilares de inversores, de quadro de proteção em corrente alternada (CA) e de *strings* FV; todas podem ser vistas na Figura 24. Para contornar a limitação da primeira versão do Revit 2023 de preencher parâmetros do tipo “Número inteiro” através do Dynamo optou-se por utilizar parâmetros intermediários do tipo “Número” nos diagramas, como pode ser visto na Figura 24.

Figura 24 – Famílias paramétricas de diagramas trifilares para projetos.



Fonte: o autor.

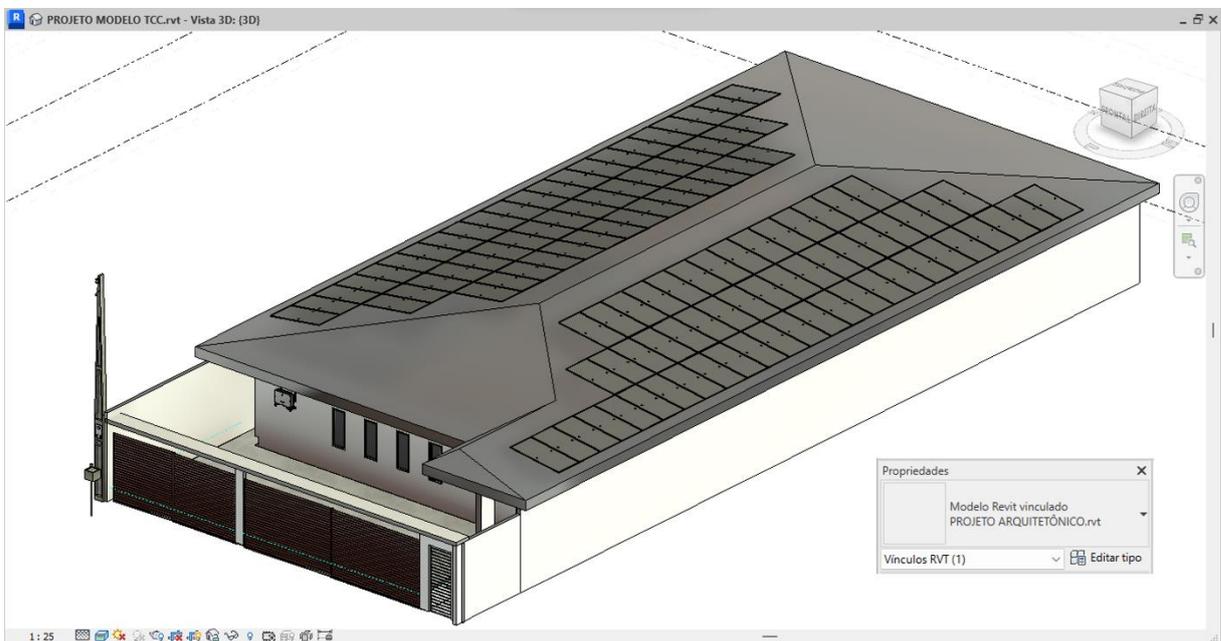
Como relatado anteriormente, dentre todas as atribuições do Dynamo, apenas algumas foram abordadas. Dessas, as principais partes encarregadas por atividades que auxiliam na elaboração de projetos serão descritas a seguir. Os resultados podem ser avaliados neste capítulo e os *scripts* completos estão disponíveis no APÊNDICE A, para a rotina de quadros, e no APÊNDICE B, para a rotina de diagramas.

O ambiente e a sequência de desenvolvimento do projeto foram definidos para atender à metodologia BIM e os procedimentos regulares adotados por empresas que usam dessa tecnologia. Sendo assim, o engenheiro eletricista deve receber o projeto estrutural/arquitetônico do local onde será instalado o sistema fotovoltaico e importá-lo como vínculo no Revit. Em seguida, a montagem dos dispositivos e equipamentos

é feita nesse modelo, respeitando seus limites e garantindo a interoperabilidade. O empreendimento vinculado que comportará o sistema fotovoltaico para o presente caso de estudo está ilustrado na Figura 25, onde os módulos FV foram dispostos sobre o telhado e o inversor instalado na parede frontal interna da residência.

O dimensionamento do sistema baseado no consumo de energia do cliente e no desempenho anual do sistema, dentre outras variáveis, não faz parte do escopo do presente trabalho e por isso não será abordado. Contudo, foi elaborado o modelo de um empreendimento padrão de microgeração distribuída apresentado na Figura 25 para atender uma residência de grande porte, com as principais características dispostas na Tabela 5.

Figura 25 – Projeto civil vinculado ao projeto elétrico já com alguns equipamentos.



Fonte: o autor.

Tabela 5 – Características do sistema solar fotovoltaico do caso em estudo.

| Atributo                  | Descrição       |
|---------------------------|-----------------|
| Potência do sistema       | 63,825 kWp      |
| Fabricante dos módulos FV | Jinko Solar     |
| Quantidade de módulos FV  | 111             |
| Modelo dos módulos FV     | JKM575N-72HL4-V |
| Potência dos módulos FV   | 575 Wp          |
| Fabricante do inversor    | Sungrow         |
| Quantidade de inversores  | 1               |

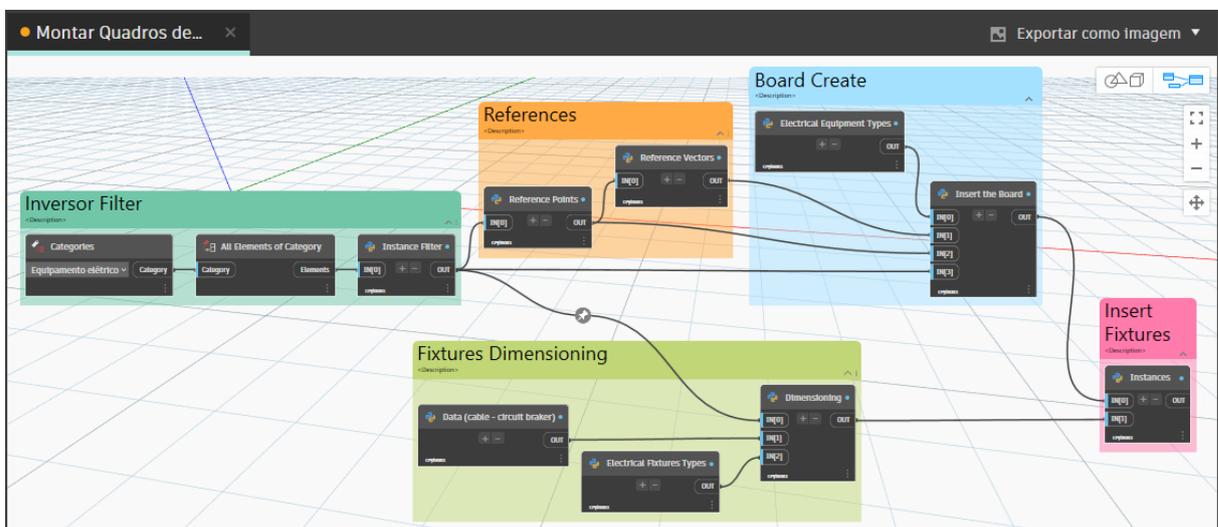
| Atributo             | Descrição |
|----------------------|-----------|
| Modelo do inversor   | SG50CX    |
| Potência do inversor | 50 kW     |

Fonte: o autor.

### 3.3 Rotina para dimensionamento do quadro elétrico

A Figura 26 apresenta a visão geral da primeira automação desenvolvida, a qual tem como proposta inicial inserir o quadro de proteção metálico na parede em que está instalado o inversor, respeitando os afastamentos mínimos exigidos no seu manual de instalação. Para tal, com os blocos do grupo “*Inversor Filter*” a rotina rastreia o equipamento no projeto.

Figura 26 – Visão geral da rotina para dimensionamento e montagem do quadro de proteção.



Fonte: o autor.

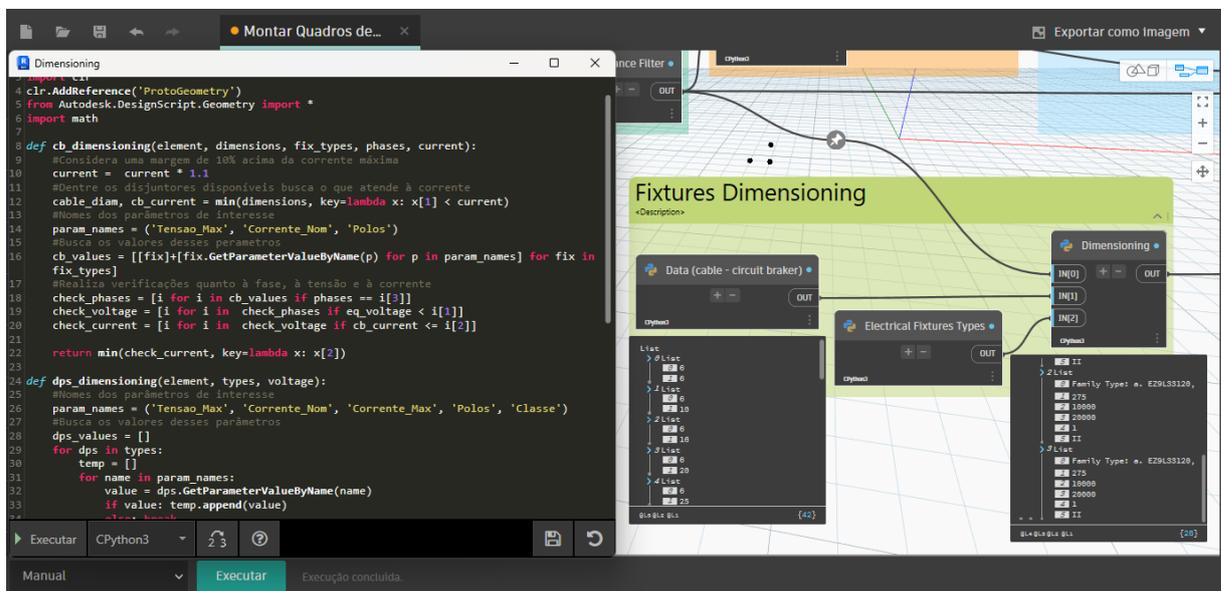
Em seguida, utilizando funções semelhantes às mostradas na Figura 11, o grupo de blocos “*References*” busca suas referências para que o Dynamo entenda os limites de sua geometria do equipamento. Dessa forma, é possível definir o ponto para inserção da família de quadro metálico com os blocos do grupo “*Board Create*”.

Em paralelo, o grupo de blocos “*Fixtures Dimensioning*” faz o dimensionamento das características do disjuntor, como o número de polos e a corrente nominal, de

acordo com os dados do inversor associado – como a corrente máxima de operação, por exemplo. Também é indicado, apesar de não chegar a ser utilizado nesta automação, o diâmetro mínimo do cabo de cobre CA que pode ser conectado entre o disjuntor de proteção da UFV e o disjuntor de conexão com a rede da distribuidora. Esse dimensionamento é associativo e válido apenas para um lance curto de cabos, uma vez que para longas distâncias faz-se necessário cálculos mais analíticos que levam em consideração outras implicações no sistema, como a queda de tensão.

A rotina também quantifica e dimensiona os DPS para proteção contra surtos causadas devido à ocorrência de descargas atmosféricas. Diferentemente do disjuntor, para o DPS algumas definições são mais subjetivas, pois não implicam unicamente na análise do sistema e sim de outras características. A corrente máxima e a classe são exemplos, pois suas escolhas levam em consideração o local e as condições em que o dispositivo será instalado. De maneira a atender a maioria dos projetos de UFVs de microgeração, optou-se por dimensionar os DPS seguindo três critérios: verificar a tensão de operação; adotar a Classe II; eleger o de menor corrente máxima dentre os disponíveis, dado que seu custo financeiro é menor. Na Figura 27 é possível observar parte do *script* responsável por realizar essas tarefas e os resultados no visualizador do bloco “Dimensioning”. O *node* “Data (cable – circuit breaker)” armazena a relação de diâmetro de cabo e seu respectivo disjuntor.

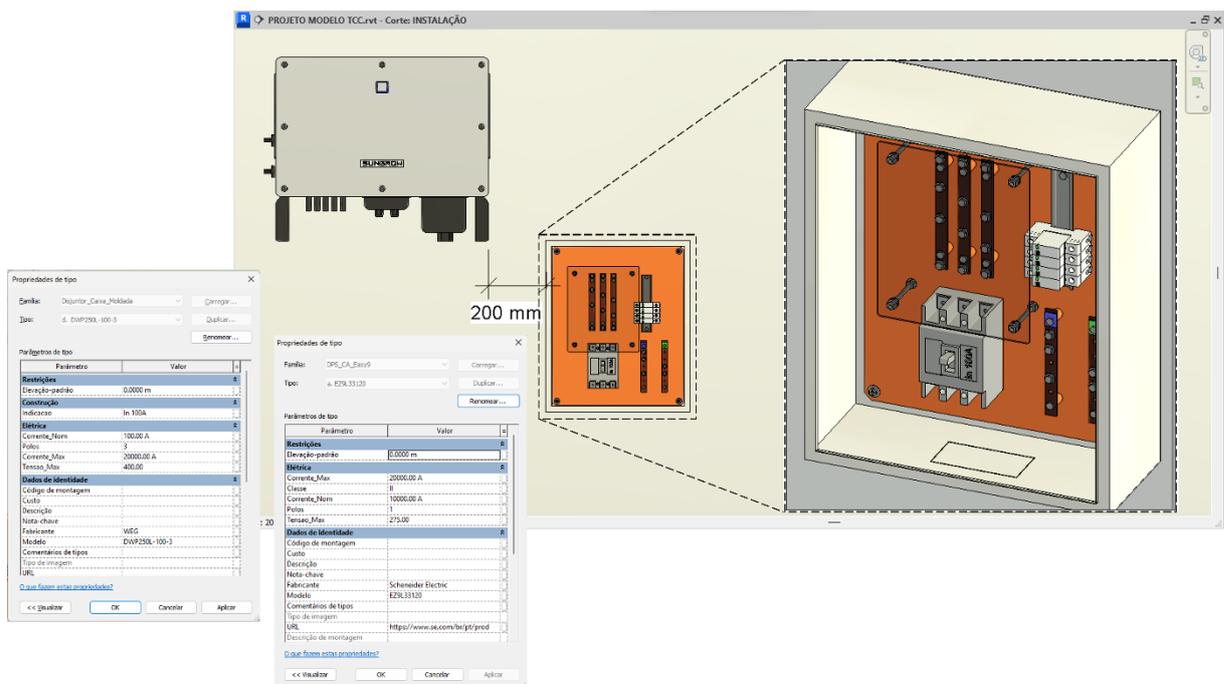
Figura 27 – Script para dimensionamento do disjuntor e dos DPS.



Por fim, detendo todas as informações necessárias, com o bloco “*Instances*” a rotina é capaz de montar o quadro elétrico na posição correta em relação ao inversor, inserir os dispositivos de proteção nos locais adequados – com o controle de seus tipos, e preencher os valores dos parâmetros elétricos que caracterizam esses elementos.

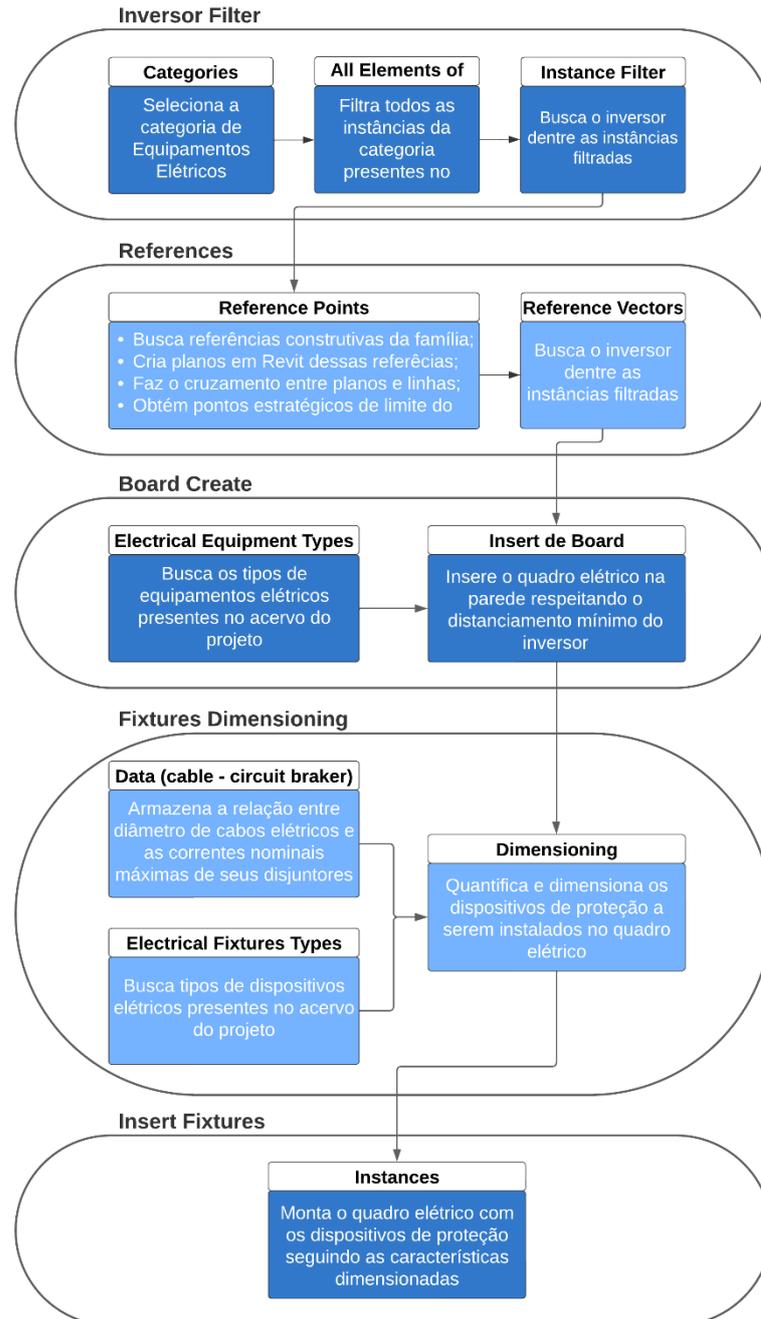
Na Figura 28 é possível observar a vista frontal do resultado gerado pela automação, bem como uma perspectiva 3D final do quadro elétrico a direita. No canto inferior esquerdo está o painel de parâmetros dos dispositivos instalados. Todos os *scripts* internos desses nodes estão descritos no APÊNDICE A e o fluxograma descritivo é apresentado na Figura 29. As atividades executadas por automação levaram cerca de 1 segundo para serem finalizadas, considerando que todas as simulações realizadas neste trabalho utilizaram de notebook com processador Intel® Core™ i5-8265 CPU @ 1.60GHz 1.80Ghz e 16 GB de RAM.

Figura 28 – Resultado gerado pela rotina de montagem do quadro de proteção.



Fonte: o autor.

Figura 29 – Fluxograma do *script* para a automação de montagem dos quadros de proteção.



Fonte: o autor.

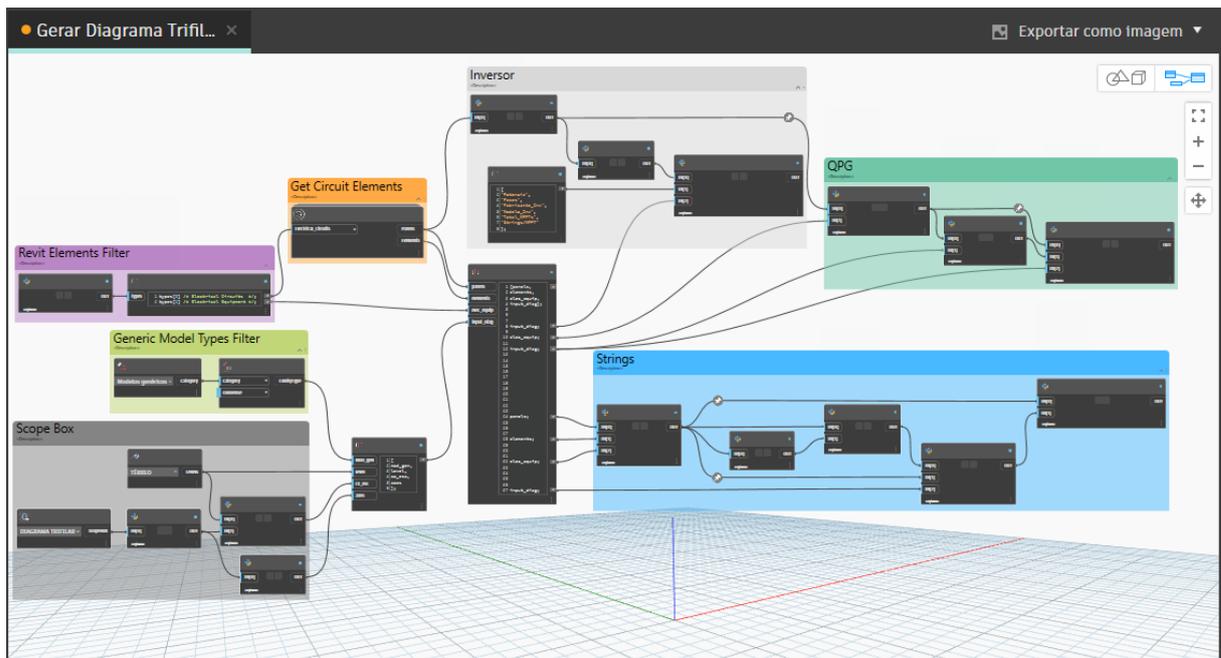
### 3.4 Rotina para criação de diagramas trifilares

Na Figura 30 está ilustrada a visão geral da rotina no Dynamo para a criação do diagrama trifilar dos dispositivos e equipamentos montados no sistema apresentado

anteriormente. De acordo com o diagrama do Dynamo, os grupos de *nodes* “Revit Elements Filter”, “Generic Model Types Filter” e “Scope Box” dão início à execução do *script*. O primeiro deles, em roxo, filtra as instâncias de equipamentos (como PDE e o inversor) e os elementos de circuito criados no Revit, que são atribuídos aos módulos FV, conectando-os ao inversor e conseqüentemente à rede de BT da distribuidora de energia.

O segundo, em verde, busca todos os tipos de famílias da categoria modelo genérico carregadas no arquivo do projeto, pois, como definido em 3.2.2, é nessa categoria que foram criados os modelos de diagramas que serão utilizados. O terceiro, em cinza, busca a caixa de escopo, criada no Revit para comportar todas as representações do diagrama, e calcula seu centro para ser o ponto de partida para a criação do diagrama trifilar. Desse modo, o primeiro diagrama criado é o do inversor, pois, a partir dele, todos os outros podem ser conectados.

Figura 30 – Visão geral da rotina de criação de diagramas trifilares.

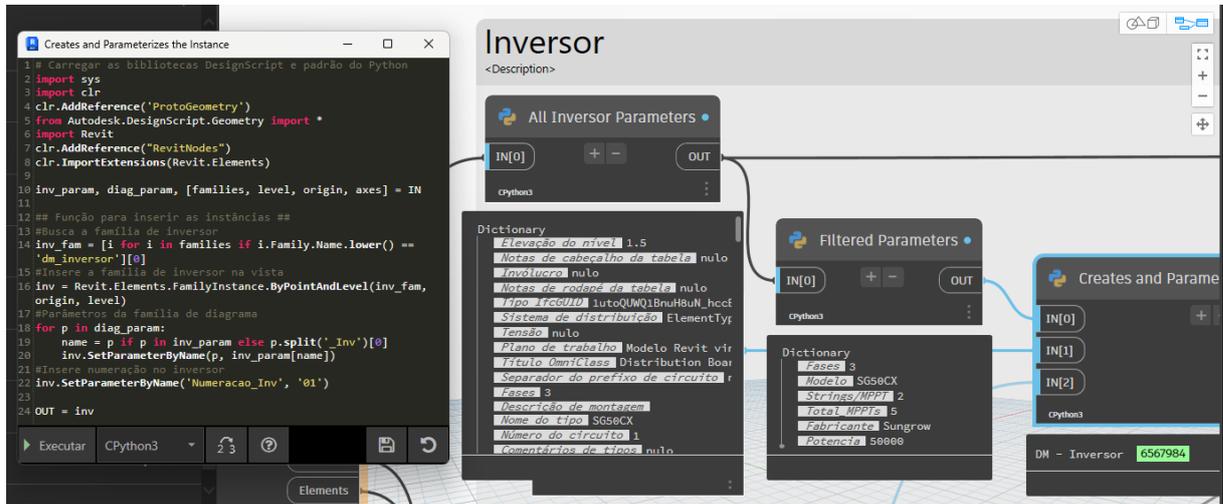


Fonte: o autor.

O bloco “*ElectricalCircuit Get Panels and Elements*”, pertencente ao pacote MEPOver, que no projeto está indicado dentro do grupo “*Get Circuit Elements*” em laranja, retorna, separadamente, uma lista de circuitos com os painéis elétricos e outra

com todos os elementos separados por circuito. No grupo “*Inversor*”, de todas as informações contidas no modelo 3D, são retiradas apenas as de interesse. Em seguida é criada a instância do diagrama e, por fim, os valores são preenchidos nos seus respectivos parâmetros, como mostra a Figura 31.

Figura 31 – Fluxograma e *script* para criação de diagrama trifilar do inversor.



Fonte: o autor.

No grupo “QPG”, em verde, a metodologia é análoga à adotada para os inversores. Primeiramente são buscados todos os dispositivos internos ao quadro de proteção metálico instalado no projeto; em seguida são filtrados apenas os parâmetros de interesse desses dispositivos e; por fim são criadas as instâncias de diagrama trifilar do quadro montado e dos cabos CA que conectam inversor e QPG.

Os nodes do grupo “*Strings*”, em azul, além de filtrarem os dados, tratá-los e criarem as instâncias de diagramas, como nos casos anteriores, fazem também análises necessárias. Essas atividades referem-se à montagem das MPPTs, pois para a conexão de *strings* diferentes em paralelo em um mesmo terminal MPPT do inversor alguns critérios devem ser seguidos, como:

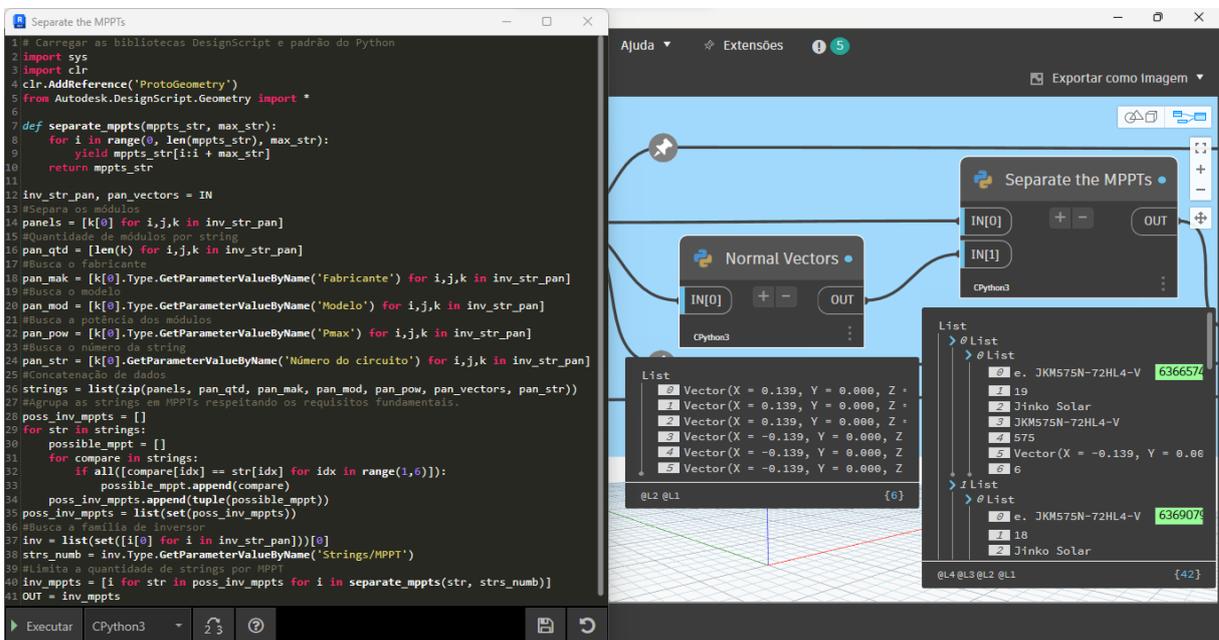
- Módulos de mesmas características elétricas, como tensão de circuito aberto (Voc), corrente de curto-circuito (Isc) e potência máxima (Pmax);
- Mesma quantidade de módulos, para garantir tensões equivalentes – já que as *strings* são montadas em série;

- c. Mesma orientação em relação ao norte (azimute) e inclinação em relação ao nível do chão, para todos os módulos.

Para tal, após separar os conjuntos de módulos para todas as *strings* do inversor, a rotina, através do bloco “*Normal Vectors*”, busca o vetor normal ao plano da face superior dos módulos de cada *string*, como pode ser visto na Figura 32. Dessa forma, é possível saber o azimute e a orientação dos módulos, permitindo então a verificação de compatibilidade e possibilidade de conexão em uma mesma MPPT.

Em seguida, no *node* “*Separate the MPPTs*” são feitas as outras compatibilizações citadas. De forma direta, optou-se por distribuir as *strings* nos dois terminais de cada MPPT garantindo que fossem montadas com a mesma quantidade de módulos. Além disso, assegura-se também que sejam equipamentos de mesmo fabricante, modelo, potência máxima e que estejam sob a mesma orientação e inclinação; seguindo assim, todos os critérios mencionados anteriormente. Na Figura 32 é possível analisar que a saída do bloco foi estruturada para a lista principal representar a lista principal das MPPTs e as terciárias suas *strings*, com todas as características de seus módulos, na ordem: instância, quantidade, fabricante, modelo, potência máxima, vetor normal e número do circuito.

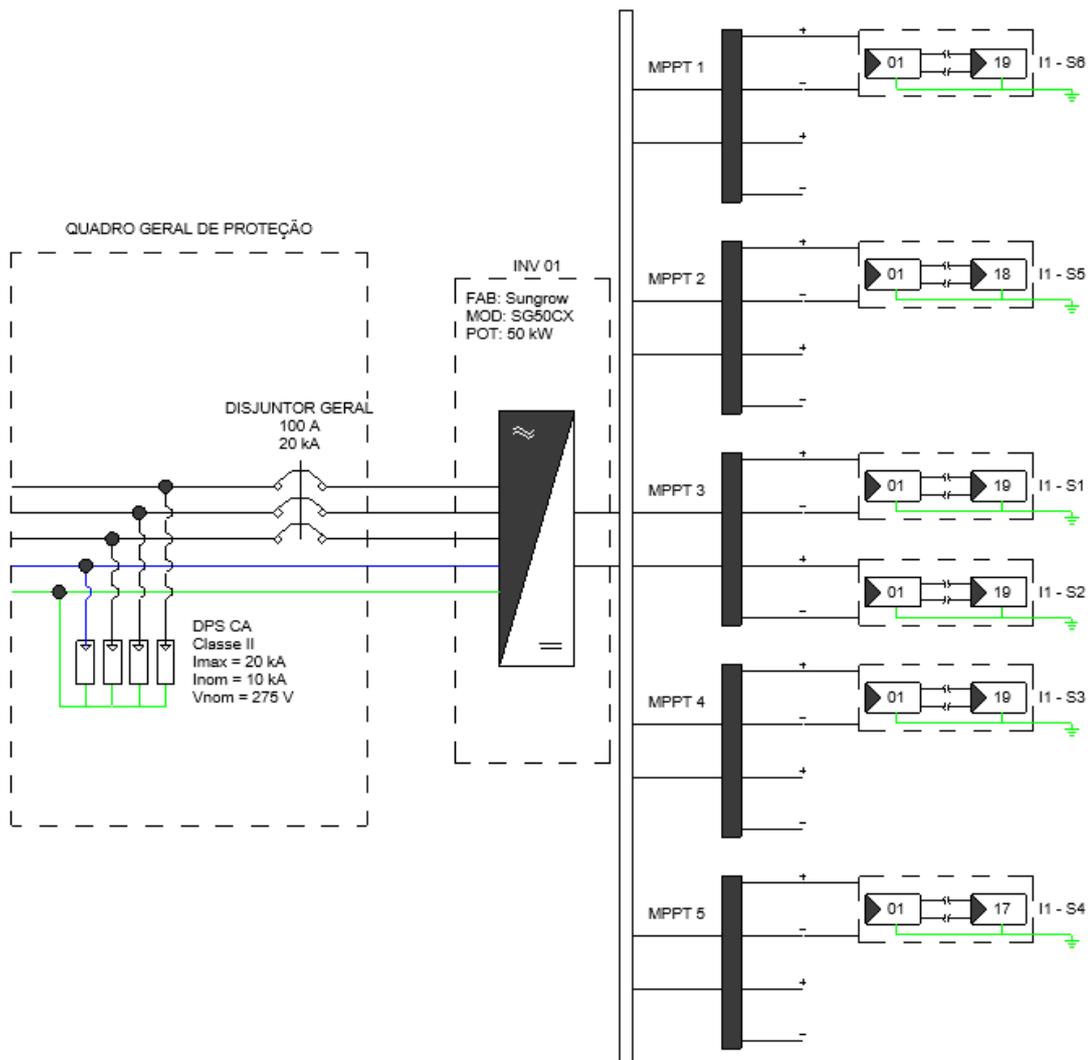
Figura 32 – *Script* para separação das strings nas MPPTs do inversor.



Fonte: o autor.

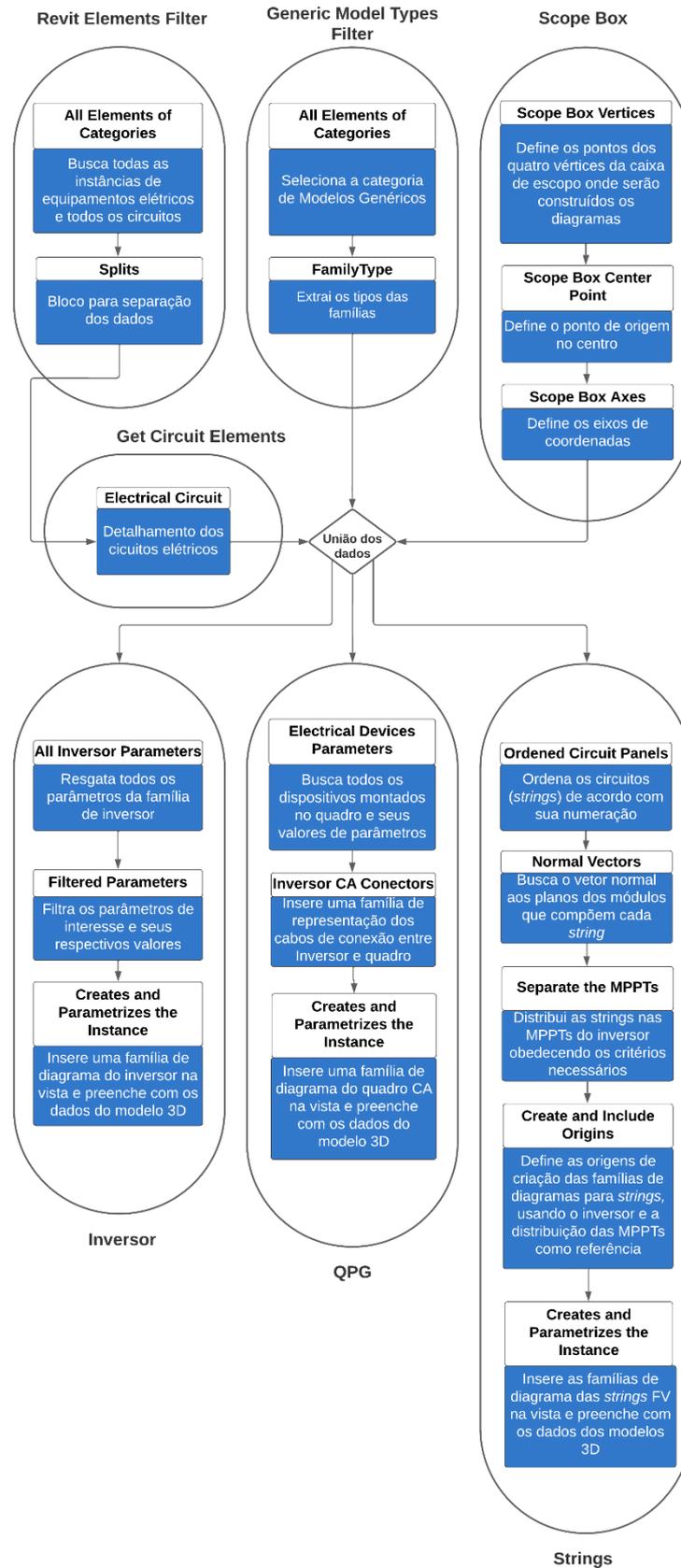
Nos últimos dois blocos do grupo são definidos os pontos de inserção de cada diagrama de *string*, de modo que se encaixem nos terminais do inversor, seguida da criação de cada instância e suas respectivas parametrizações. Assim como na seção 3.3 automação leva cerca 1,5 segundos para finalizar, utilizando o mesmo *hardware*, e seus resultados são apresentados na Figura 33. Todos os *scripts* internos dos nodes para a criação do diagrama trifilar estão descritos no APÊNDICE B e o fluxograma dos *scripts* pode ser visto na Figura 34.

Figura 33 – Resultado gerado pela rotina de criação de diagramas trifilares.



Fonte: o autor.

Figura 34 – Fluxograma do *script* para a automação de criação de diagramas trifilares.



Fonte: o autor.

## 4 CONCLUSÕES E PROPOSTAS DE CONTINUIDADE

Com este trabalho foi possível atestar algumas das muitas aplicabilidades da metodologia BIM, com destaque para a execução de projetos elétricos de usinas solares fotovoltaicas. Foi possível conferir certas funcionalidades do *software* Revit que atendem às necessidades de modelagem de circuitos e equipamentos elétricos de um sistema, enfatizando na construção de informações sólidas que são transferidas e usadas em situações diferentes de um mesmo projeto, unificando a fonte de dados e garantindo confiabilidade ao projeto. Essa perspectiva justifica a abordagem cuidadosa no presente trabalho para com a modelagem dos elementos e seus parâmetros que, apesar de demandar mais tempo inicialmente, proporciona celeridade e praticidade a posteriori.

Além do mais, o foco numa abordagem mais minuciosa na construção inicial do sistema no Revit objetivou uma interoperabilidade sólida com o Dynamo, que necessita do ambiente bem estruturado para que as automações executadas por intermédio dele atuem corretamente. Com as rotinas desenvolvidas buscou-se dimensionar dispositivos elétricos, realizar instalações e elaborar montagens de forma automática, seguindo as normas vigentes e critérios definidos pelos fabricantes. Portanto, todas essas atividades e cálculos que seriam realizadas manualmente pelo projetista passam a ser executadas pelos *scripts*, reduzindo a poucos segundos uma atividade que levaria horas para ser finalizada, a depender do tamanho do sistema FV projetado.

Como proposta de aprimoramento para o trabalho aqui desenvolvido, enxerga-se a ampliação da abrangência das automações, objetivando atender sistemas solares de minigeração que, normalmente, são compostos por vários inversores e milhares de módulos. Para esse alcance é necessário o dimensionamento e montagem de mais dispositivos elétricos no quadro de proteção 3D e do tratamento mais aprimorado na rotina de diagramas. Para esse segundo caso, as famílias modeladas de diagramas já possuem autonomia suficiente e podem ser aplicadas. Como continuidade iminente aos resultados já alcançados, propõe-se utilizar a verificação de azimute e inclinação dos módulos para identificar montagens incorretas e as informações das *strings* para acusar a violação dos limites suportados de tensão e corrente nos terminais MPPTs do inversor.

## REFERÊNCIAS

1. CREDER, H. **Instalações Elétricas**. 16ª. ed. Rio de Janeiro: LTC, 2016. ISBN 978-85-216-3072-2.
2. DE OLIVEIRA, A. M.; MARIO, M. C.; PACHECO, M. T. T. Fontes renováveis de energia elétrica: evolução da oferta de energia fotovoltaica no Brasil até 2050. **Brazilian Applied Science Review**, Curitiba, v. 5, n. 1, p. 257-272, Janeiro 2021.
3. SANHUDO, L. et al. A digitalização nas empresas da Indústria AEC portuguesa: Inquérito e análise de resultados, p. 13-17, Outubro 2022.
4. SCHIMITT, V. C.; ALONSO, E. E. D. M. Aplicativo em plataforma CAD para auxílio no desenho de projetos elétricos em baixa tensão. **3rd International Conference on Engineering and Computer Education**, São Paulo, p. 5, Março 2003.
5. EASTMAN, C. et al. **Manual de BIM: um guia de modelagem da informação da construção para arquitetos, engenheiros, gerentes, construtores e incorporadores**. 1ª. ed. Porto Alegre: Bookman, 2014. 483 p.
6. FAROOQ, J.; SHARMA, P.; R, S. K. Applications of Building Information Modeling in Electrical Systems Design. **Journal of Engineering Science and Technology Review**, v. 10, n. 6, p. 119-128, Dezembro 2017.
7. FERREIRA, A. R. et al. Revisão da literatura: uso do conceito BIM em projetos do setor elétrico nos cenários (Inter)Nacional. **Research, Society and Development**, v. 11, n. 6, p. 34, Abril 2022.
8. THÓRUS ENGENHARIA. **Cenário Construtivo Brasileiro**. Thórus Engenharia. [S.l.], p. 1-61. 2019.
9. SUCCAR, B. Building information modelling framework: A research and delivery foundation for industry stakeholders. **Automation in construction**, v. 18, n. 3, p. 357-375, 2009.
10. MONTEIRO, A.; MARTINS, J. P. **Building Information Modeling (BIM) - teoria e aplicação**. International Conference on Engineering UBI. Covilhã: [s.n.]. 2011. p. 1-10.
11. DE SENA, T. S. A aplicação da metodologia BIM para a compatibilização de projetos. **Escola Politécnica, Universidade Federal da Bahia**, Salvador, 2012.
12. MACHADO, I. D. C. G. Automatização de Processos BIM através de Dynamo aplicado ao Projeto de Estruturas. **Faculdade de Engenharia da Universidade do Porto**, Porto, Junho 2022.
13. SECRETARIA-GERAL DA PRESIDÊNCIA DA REPÚBLICA. Decreto nº 9.377, de 17 de maio de 2018. **Portal da Legislação**, 17 Maio 2018. Disponível em: <[https://www.planalto.gov.br/ccivil\\_03/\\_Ato2015-2018/2018/Decreto/D9377.htm](https://www.planalto.gov.br/ccivil_03/_Ato2015-2018/2018/Decreto/D9377.htm)>. Acesso em: 4 Julho 2023.
14. SECRETARIA-GERAL DA PRESIDÊNCIA DA REPÚBLICA. Decreto nº 9.983, de 22 de agosto de 2019. **Portal da Legislação**, 22 Agosto 2019. Disponível em: <[https://www.planalto.gov.br/ccivil\\_03/\\_Ato2019-2022/2019/Decreto/D9983.htm#art15](https://www.planalto.gov.br/ccivil_03/_Ato2019-2022/2019/Decreto/D9983.htm#art15)>. Acesso em: 4 Julho 2023.
15. SECRETARIA-GERAL DA PRESIDÊNCIA DA REPÚBLICA. Decreto nº 10.306, de 2 de abril de 2020. **Portal da Legislação**, 2 Abril 2020. Disponível em:

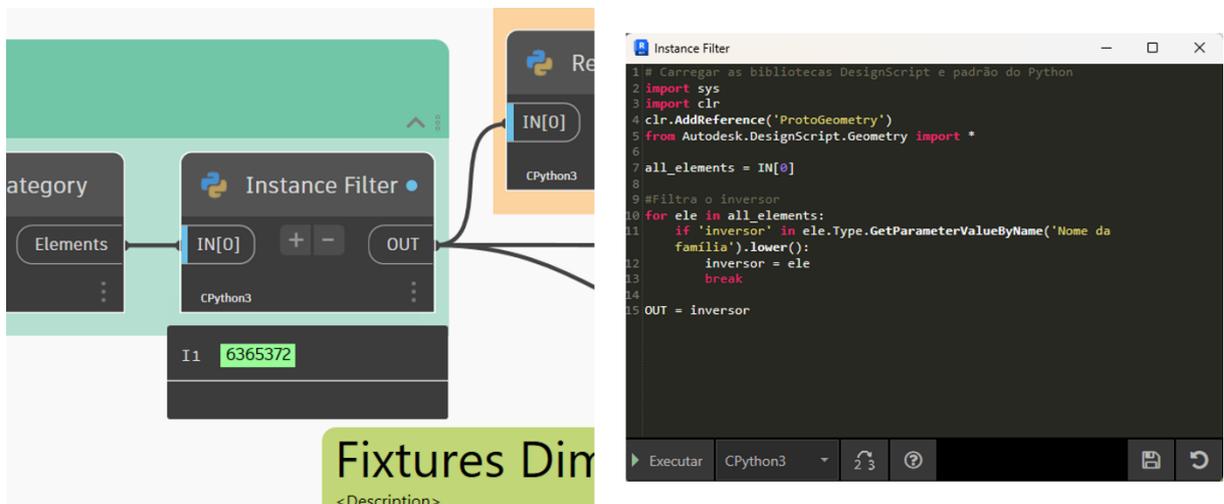
<[https://www.planalto.gov.br/ccivil\\_03/\\_ato2019-2022/2020/decreto/d10306.htm](https://www.planalto.gov.br/ccivil_03/_ato2019-2022/2020/decreto/d10306.htm)>.  
Acesso em: 4 Julho 2023.

16. ABSOLAR - ASSOCIAÇÃO BRASILEIRA DE ENERGIA SOLAR FOTOVOLTAICA. Energia Solar Fotovoltaica no Brasil, 07 Julho 2023. Disponível em: <<https://www.absolar.org.br/mercado/infografico/>>. Acesso em: 14 Julho 2023.
17. PINTO, R. F. F. Aplicação da Metodologia BIM ao Projeto de Pontes - Caso Prático. **Instituto Superior de Engenharia do Porto**, Porto, Novembro 2016.
18. AUTODESK. Online Documentation for the Revit API. **Revit API Docs**, 2019-2024. Disponível em: <<https://www.revitapidocs.com/>>. Acesso em: 23 jul. 2023.

## APÊNDICES

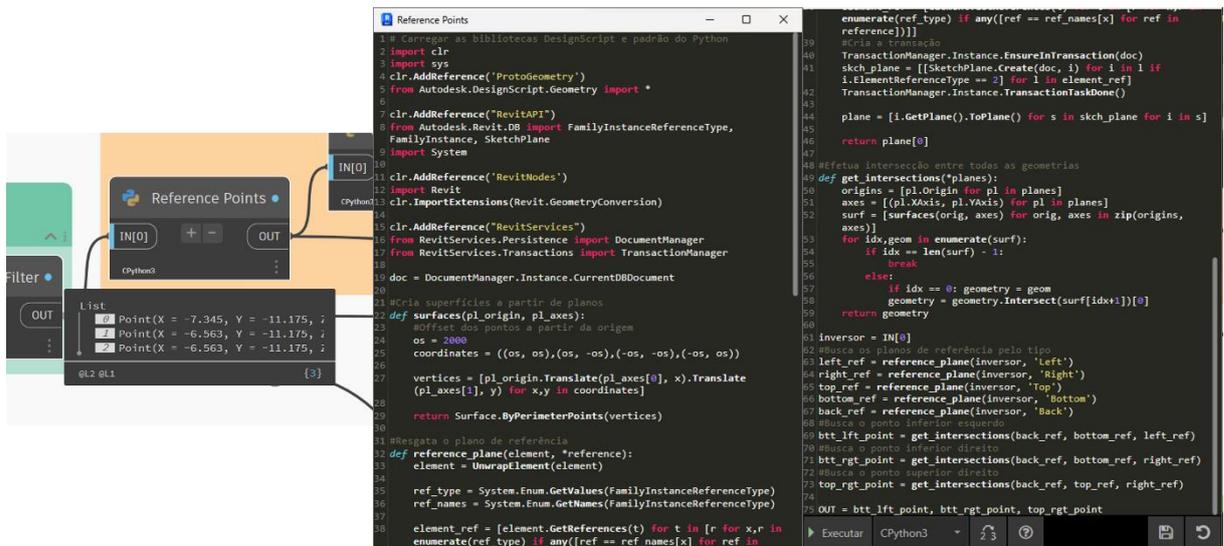
### APÊNDICE A – SCRIPTS DA ROTINA DE MONTAGEM DO QUADRO ELÉTRICO

Figura 35 – Script completo e resultados para busca de instância do inversor FV.



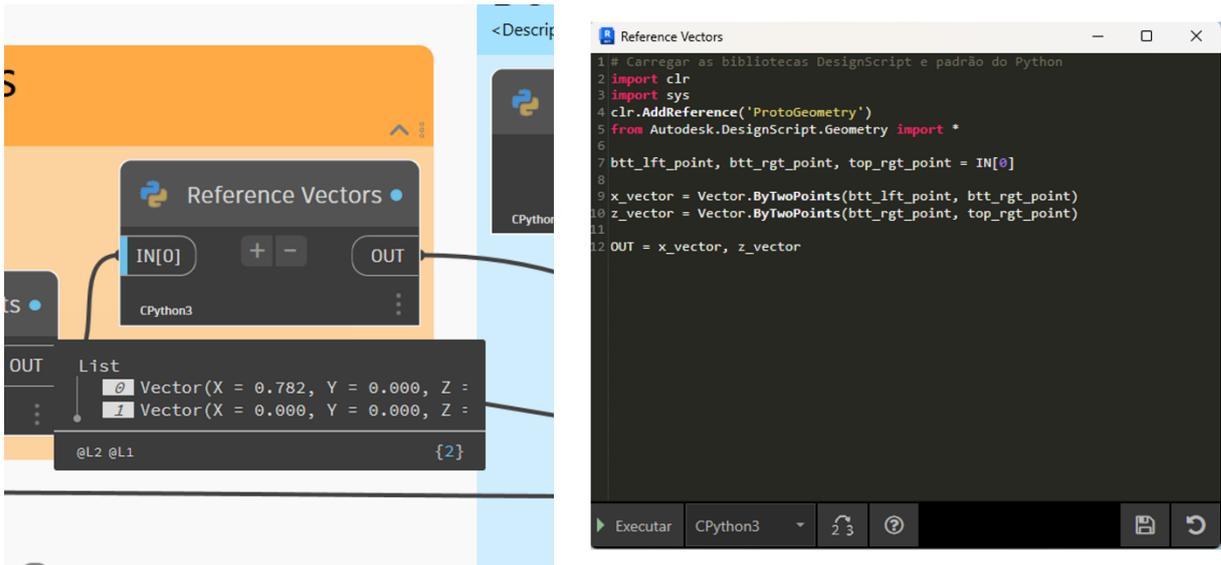
Fonte: o autor.

Figura 36 – Script completo e resultados para busca de pontos do inversor.



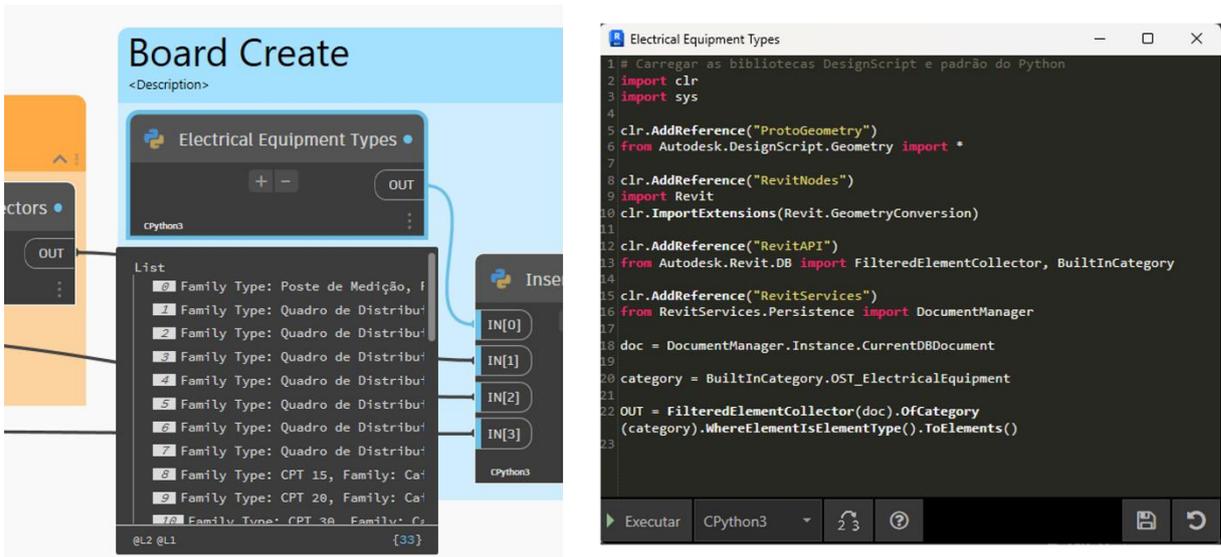
Fonte: o autor.

Figura 37 – Script completo e resultados para definição de vetores do inversor.



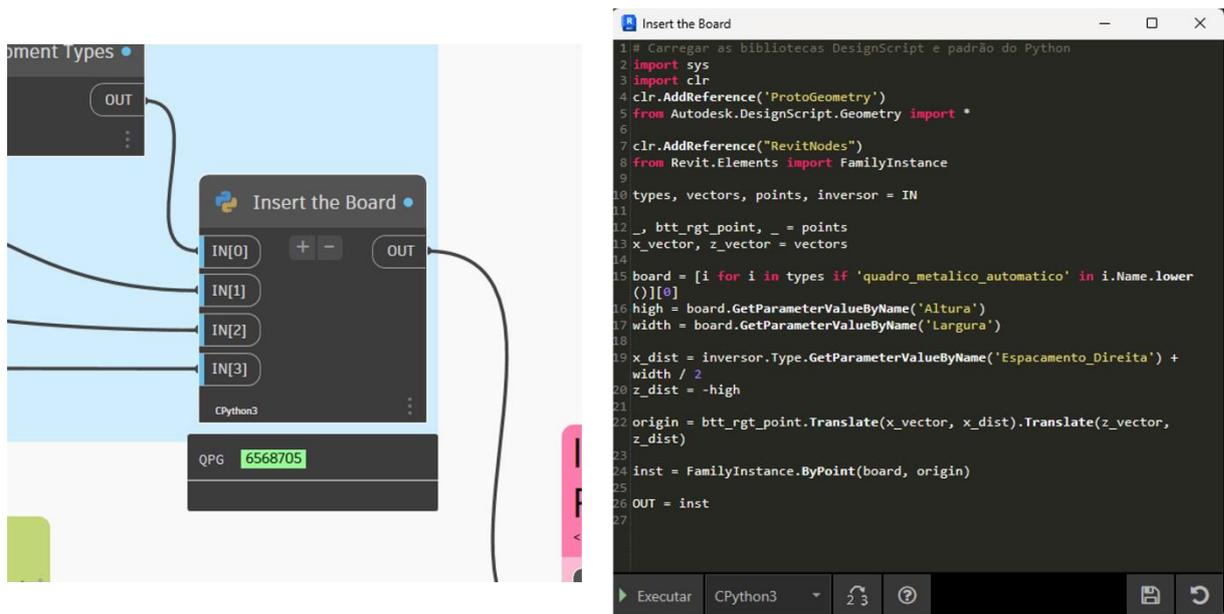
Fonte: o autor.

Figura 38 – Script completo e resultados para filtragem de equipamentos elétricos.



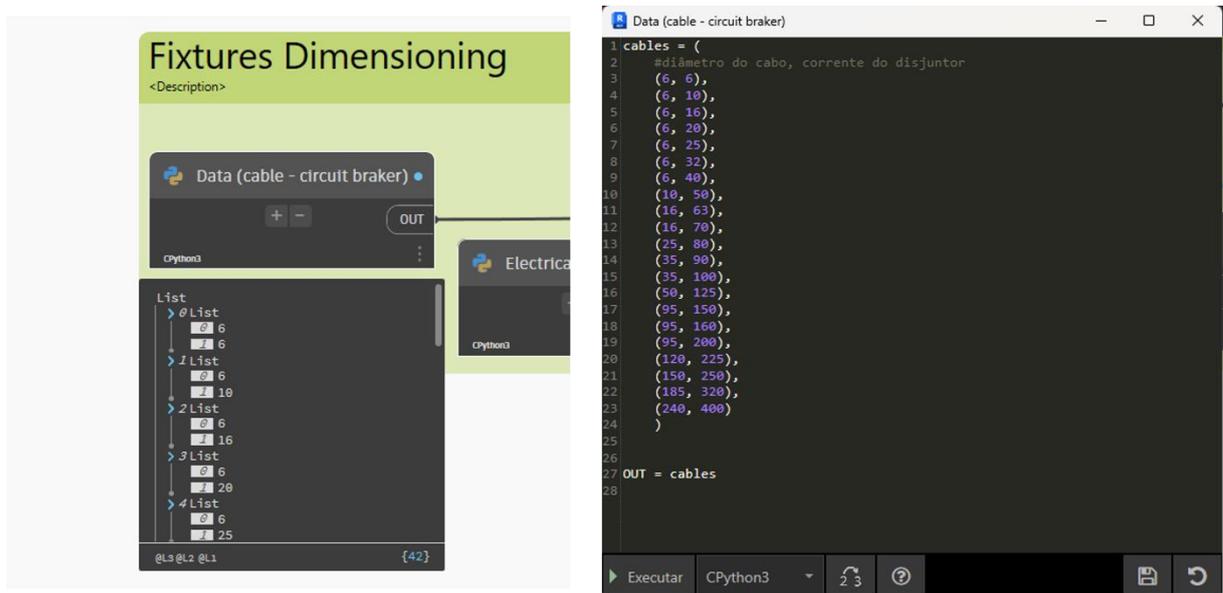
Fonte: o autor.

Figura 39 – Script completo e resultados para inserir instância de quadro metálico.



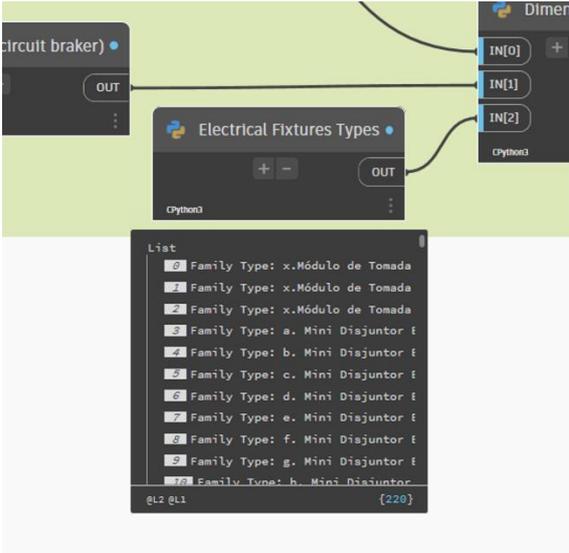
Fonte: o autor.

Figura 40 – Script com dados de cabo e seu respectivo disjuntor.



Fonte: o autor.

Figura 41 – Script completo e resultados para filtragem de dispositivos elétricos.



The screenshot shows a Dynamo script node 'Electrical Fixtures Types' with three input ports (IN[0], IN[1], IN[2]) and one output port (OUT). The results list shows the following items:

- Family Type: x. Módulo de Tomada
- Family Type: x. Módulo de Tomada
- Family Type: x. Módulo de Tomada
- Family Type: a. Mini Disjuntor
- Family Type: b. Mini Disjuntor
- Family Type: c. Mini Disjuntor
- Family Type: d. Mini Disjuntor
- Family Type: e. Mini Disjuntor
- Family Type: f. Mini Disjuntor
- Family Type: g. Mini Disjuntor
- Family Type: h. Mini Disjuntor

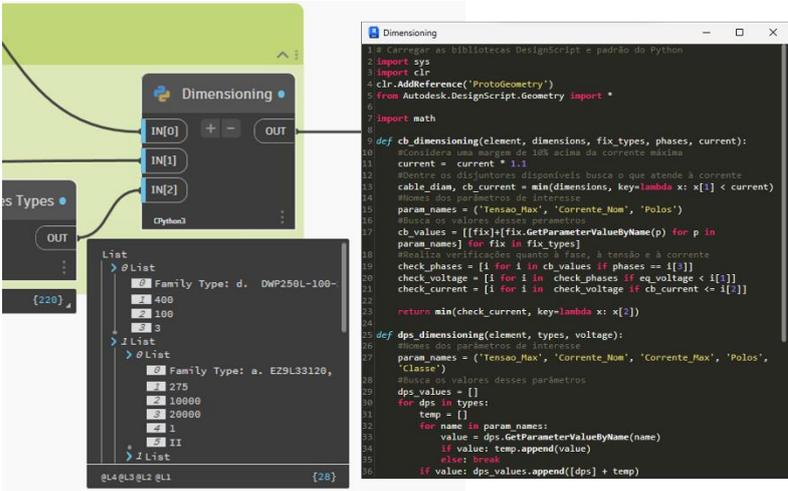
The script code in the background is as follows:

```

1 # Carregar as bibliotecas DesignScript e padrão do Python
2 import clr
3 import sys
4
5 clr.AddReference("ProtoGeometry")
6 from Autodesk.DesignScript.Geometry import *
7
8 clr.AddReference("RevitNodes")
9 import Revit
10 clr.ImportExtensions(Revit.GeometryConversion)
11
12 clr.AddReference("RevitAPI")
13 from Autodesk.Revit.DB import FilteredElementCollector, BuiltInCategory
14
15 clr.AddReference("RevitServices")
16 from RevitServices.Persistence import DocumentManager
17
18 doc = DocumentManager.Instance.CurrentDBDocument
19
20 category = BuiltInCategory.OST_ElectricalFixtures
21
22 OUT = FilteredElementCollector(doc).OfCategory
23 (category).WhereElementIsElementType().ToElements()
  
```

Fonte: o autor.

Figura 42 – Script completo e resultados para dimensionamento de dispositivos.



The screenshot shows a Dynamo script node 'Dimensioning' with three input ports (IN[0], IN[1], IN[2]) and one output port (OUT). The results list shows the following items:

- Family Type: d. DWP250L-100-
- 400
- 100
- 3
- Family Type: a. E29L33120,
- 275
- 10000
- 20000
- 1
- II

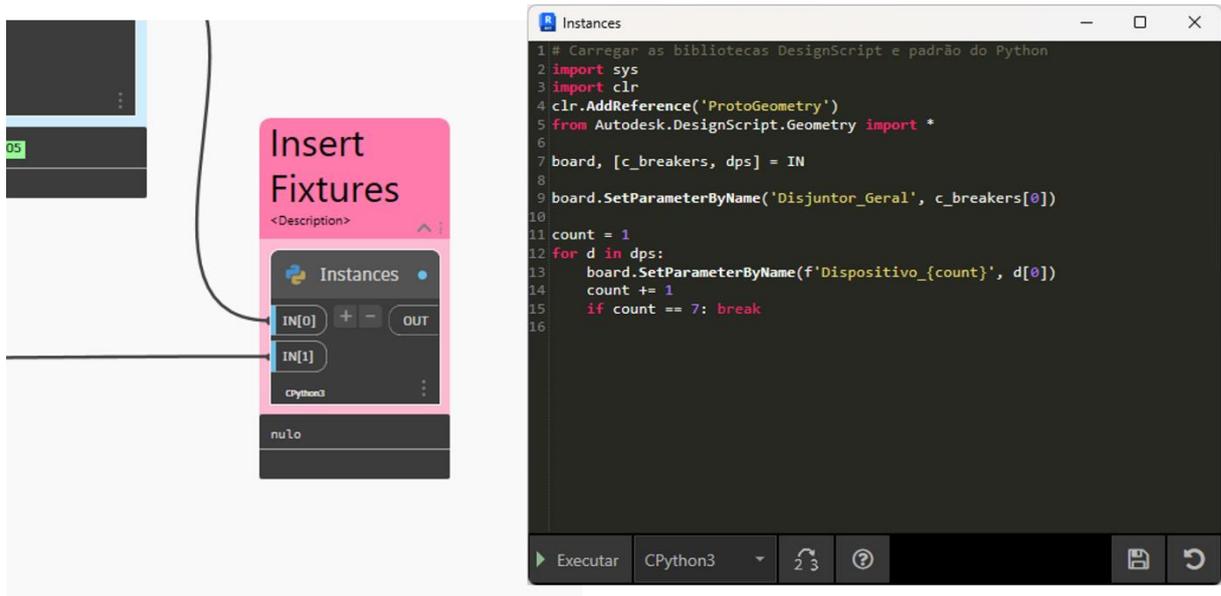
The script code in the background is as follows:

```

1 # Carregar as bibliotecas DesignScript e padrão do Python
2 import sys
3 import clr
4 clr.AddReference("ProtoGeometry")
5 from Autodesk.DesignScript.Geometry import *
6
7 import math
8
9 def cb_dimensioning(element, dimensions, fix_types, phases, current):
10     #Calculadora em função de 10x acima da corrente máxima
11     current = current * 1.1
12     #Dentro dos disjuntores disponíveis busca o que atende à corrente
13     cable_diam, cb_current = min(dimensions, key=lambda x: x[1] < current)
14     #Busca os parâmetros de interesse
15     param_names = ('Tensao_Max', 'Corrente_Nom', 'Polos')
16     #Busca os valores desses parâmetros
17     cb_values = [{"fix": fix.GetParameterByName(p) for p in param_names} for fix in fix_types]
18     #Realiza verificações quanto à fase, à tensão e à corrente
19     check_phases = [1 for i in cb_values if phases == i[1]]
20     check_voltage = [1 for i in check_phases if eq_voltage < i[1]]
21     check_current = [1 for i in check_voltage if cb_current <= i[2]]
22
23     return min(check_current, key=lambda x: x[2])
24
25 def dps_dimensioning(element, types, voltage):
26     #Busca os parâmetros de interesse
27     param_names = ('Tensao_Max', 'Corrente_Nom', 'Corrente_Max', 'Polos',
28     'Classe')
29     dps_values = []
30     for dps in types:
31         temp = []
32         for name in param_names:
33             value = dps.GetParameterByName(name)
34             if value: temp.append(value)
35             else: break
36             if value: dps_values.append([dps] + temp)
  
```

Fonte: o autor.

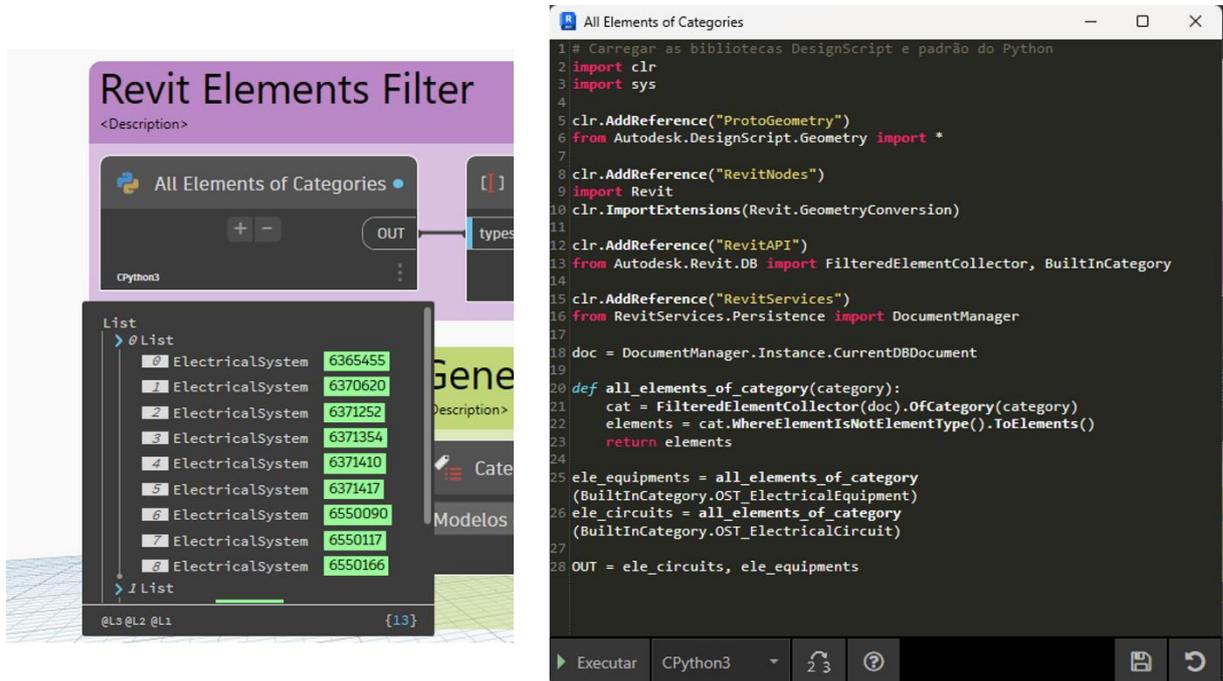
Figura 43 – Script completo para parametrização dos dispositivos.



Fonte: o autor.

## APÊNDICE B – SCRIPTS DA ROTINA DE DIAGRAMA TRIFILAR

Figura 44 – Script completo e resultados para busca de equipamentos e circuitos elétricos.



The image shows two windows from a Revit environment. On the left is the 'Revit Elements Filter' window, which displays a list of 8 'ElectricalSystem' elements with their IDs highlighted in green: 6365455, 6370620, 6371252, 6371354, 6371410, 6371417, 6550090, and 6550117. On the right is a Python script window titled 'All Elements of Categories'. The script is written in Python and uses the Revit API to filter elements. The code includes imports for clr, sys, and Revit, and defines a function 'all\_elements\_of\_category' that filters elements based on their category. The script then uses this function to retrieve 'ele equipments' and 'ele circuits' and outputs them to the 'OUT' parameter of the filter.

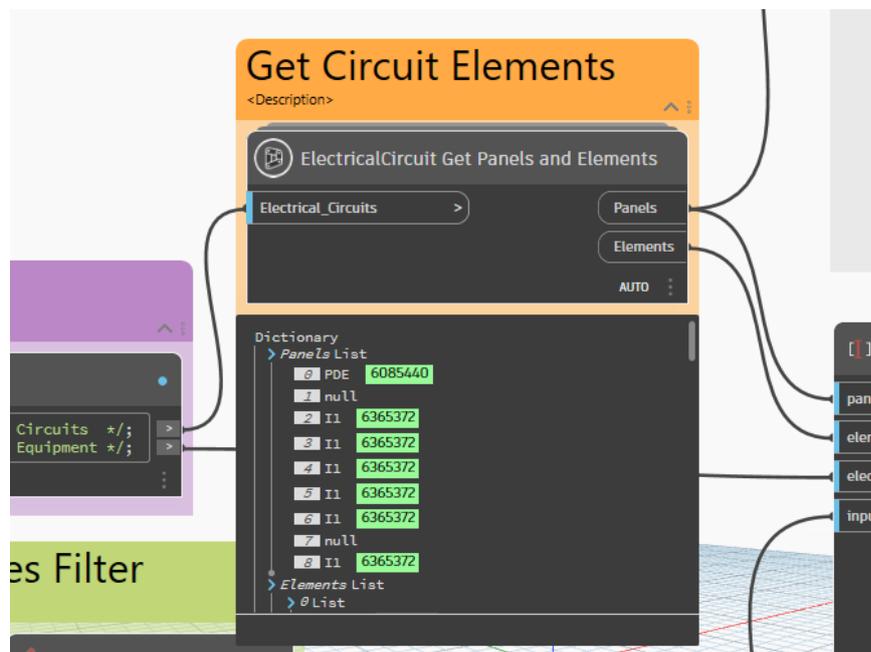
```

1 # Carregar as bibliotecas DesignScript e padrão do Python
2 import clr
3 import sys
4
5 clr.AddReference("ProtoGeometry")
6 from Autodesk.DesignScript.Geometry import *
7
8 clr.AddReference("RevitNodes")
9 import Revit
10 clr.ImportExtensions(Revit.GeometryConversion)
11
12 clr.AddReference("RevitAPI")
13 from Autodesk.Revit.DB import FilteredElementCollector, BuiltInCategory
14
15 clr.AddReference("RevitServices")
16 from RevitServices.Persistence import DocumentManager
17
18 doc = DocumentManager.Instance.CurrentDBDocument
19
20 def all_elements_of_category(category):
21     cat = FilteredElementCollector(doc).OfCategory(category)
22     elements = cat.WhereElementIsNotElementType().ToElements()
23     return elements
24
25 ele equipments = all_elements_of_category
26                 (BuiltInCategory.OST_ElectricalEquipment)
27 ele circuits = all_elements_of_category
28                (BuiltInCategory.OST_ElectricalCircuit)
29
30 OUT = ele equipments, ele equipments

```

Fonte: o autor.

Figura 45 – Bloco que separa elementos dos circuitos nos painéis elétricos.



The image shows a Revit block titled 'Get Circuit Elements'. The block has a dropdown menu set to 'Electrical\_Circuits' and buttons for 'Panels', 'Elements', and 'AUTO'. Below the block is a dictionary window showing the results of the filter. The dictionary has two main keys: 'Panels List' and 'Elements List'. The 'Panels List' contains 8 items: PDE (6085440), null, I1 (6365372), I1 (6365372), I1 (6365372), I1 (6365372), null, and I1 (6365372). The 'Elements List' is currently empty.

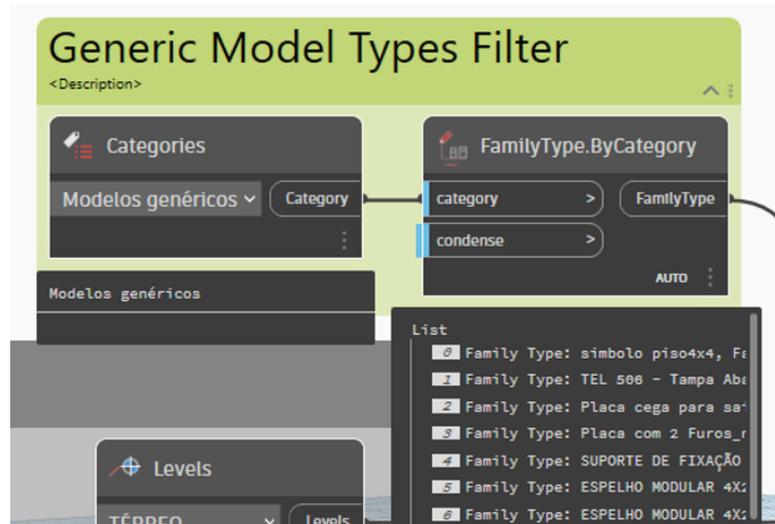
```

Dictionary
> Panels List
  0 PDE 6085440
  1 null
  2 I1 6365372
  3 I1 6365372
  4 I1 6365372
  5 I1 6365372
  6 I1 6365372
  7 null
  8 I1 6365372
> Elements List
  > List

```

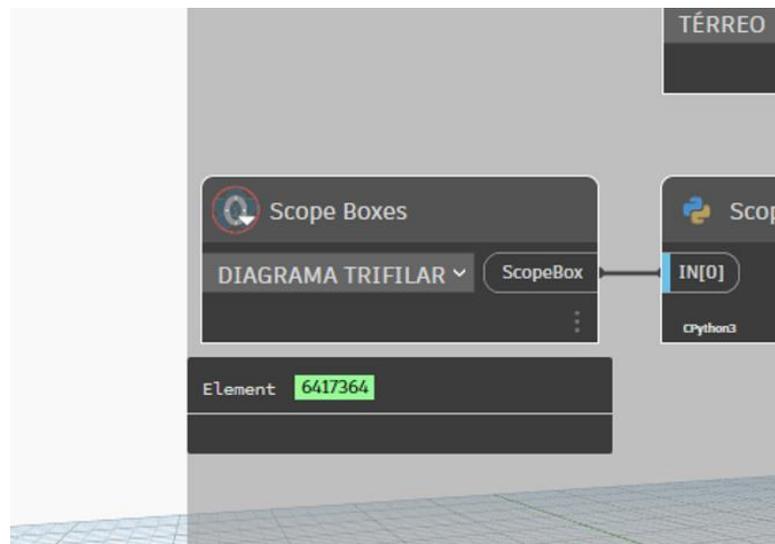
Fonte: o autor.

Figura 46 – Blocos para busca de todos os tipos da categoria Modelos genéricos.



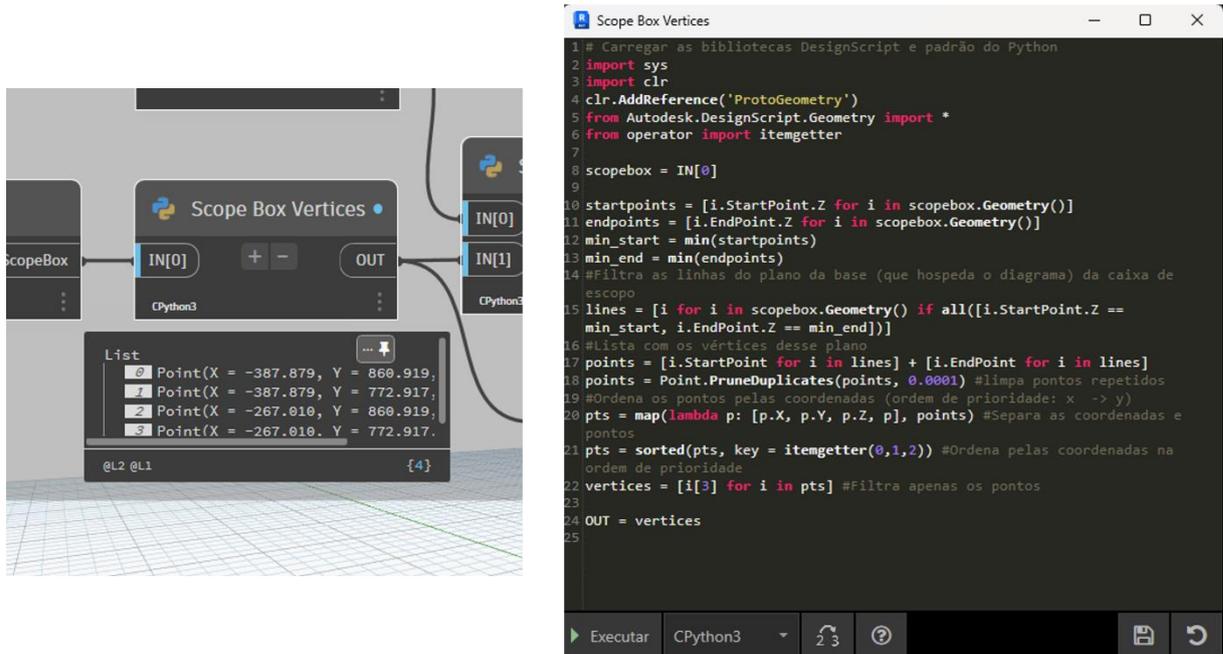
Fonte: o autor.

Figura 47 – Bloco para seleção da caixa de escopo.



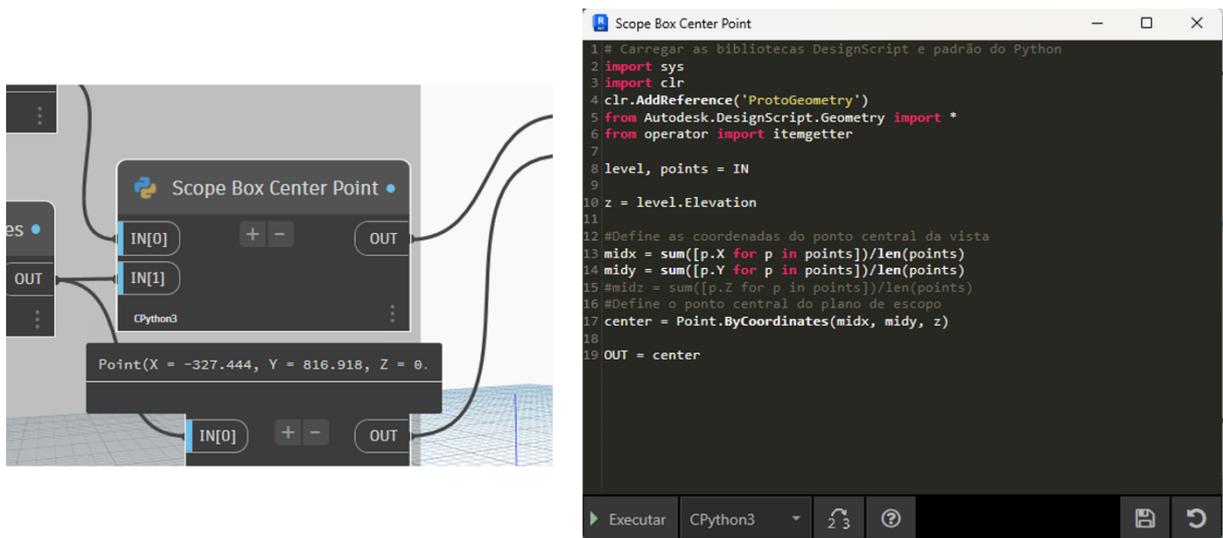
Fonte: o autor.

Figura 48 – Script completo e resultados para determinação dos vértices da caixa de escopo.



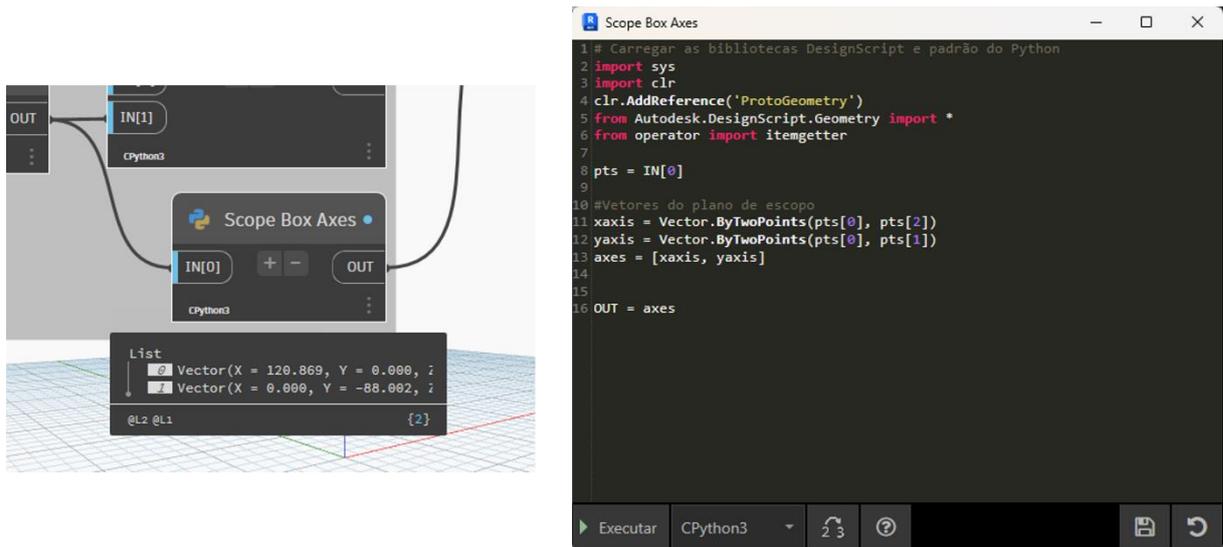
Fonte: o autor.

Figura 49 – Script completo e resultados determinação da origem do diagrama trifilar.



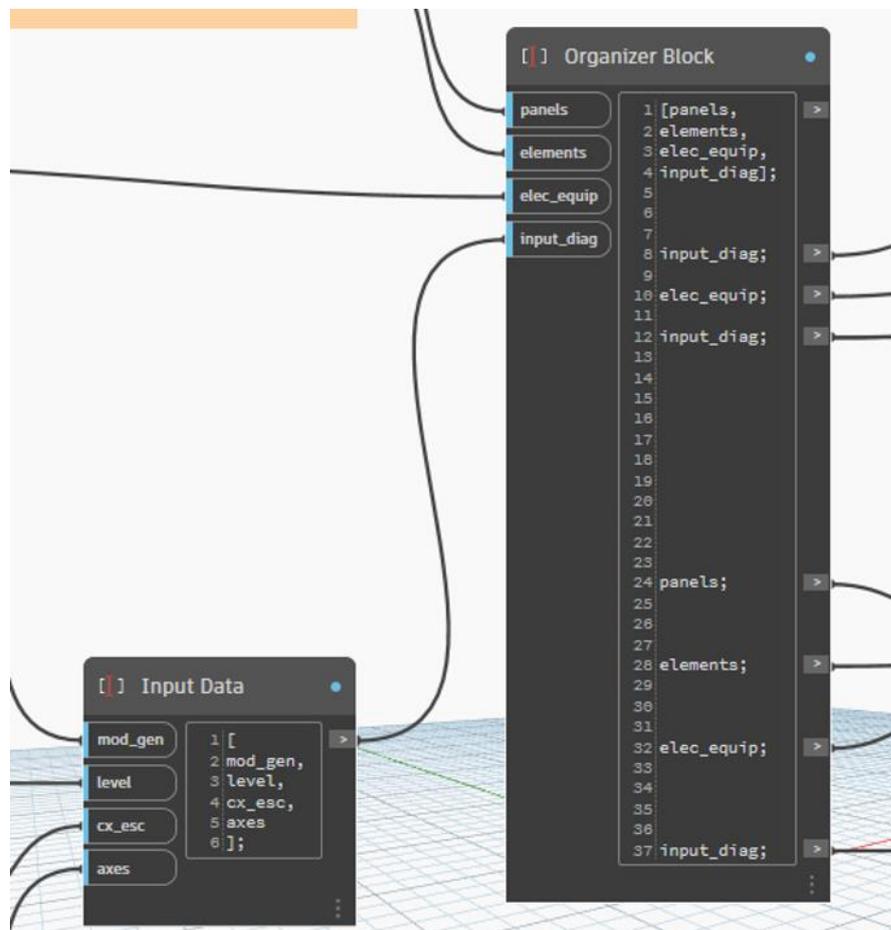
Fonte: o autor.

Figura 50 – Script completo e resultados para determinação de vetores da caixa de escopo.



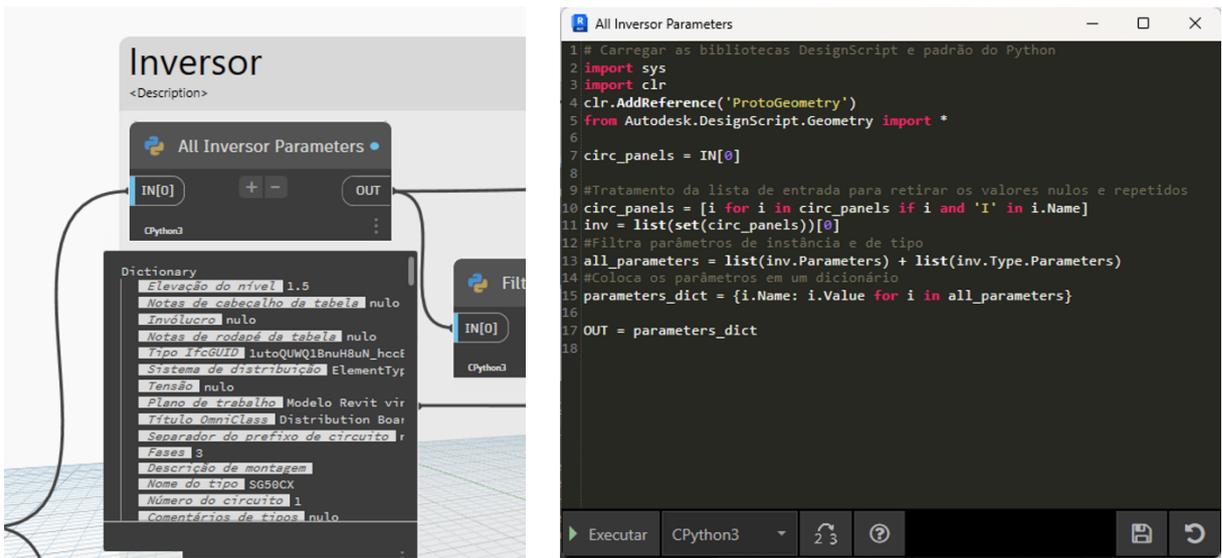
Fonte: o autor.

Figura 51 – Bloco para a organização dos fios de comunicação entre blocos.



Fonte: o autor.

Figura 52 – Script completo e resultados para busca de todos os parâmetros do inversor.



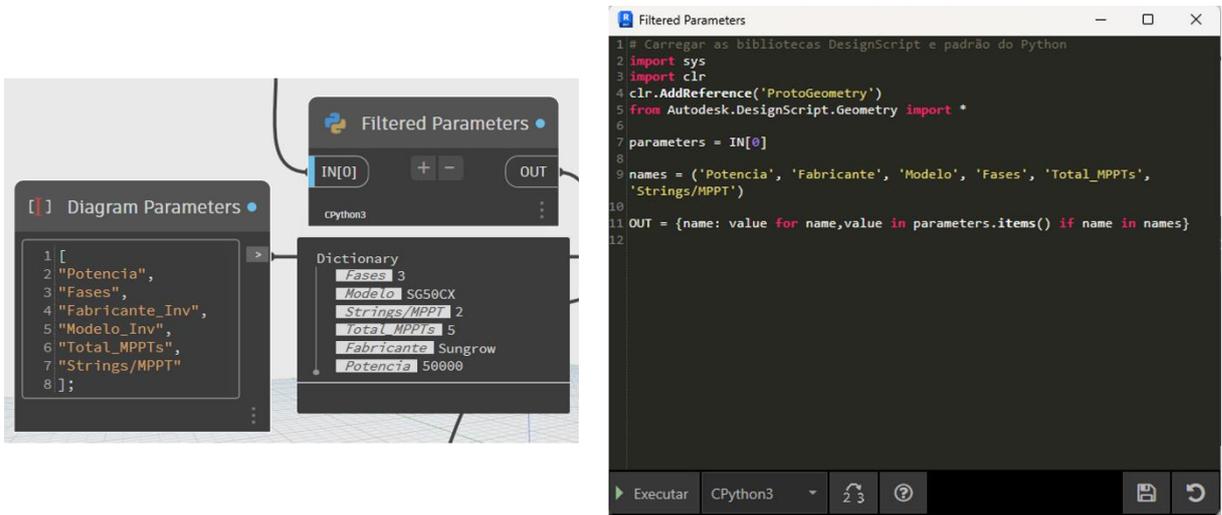
The image shows a Revit script execution interface. On the left, a 'Dictionary' window displays the output of the 'All Inversor Parameters' script. The dictionary contains various parameters such as 'Elevação do nível', 'Notas de cabeçalho da tabela', 'Invólucro', 'Notas de rodapé da tabela', 'Tipo IfcGUID', 'Sistema de distribuição', 'Tensão', 'Plano de trabalho', 'Título OmniClass', 'Separador do prefixo de circuito', 'Fases', 'Descrição de montagem', 'Nome do tipo', 'Número do circuito', and 'Comentários de tipos'. On the right, a code editor window titled 'All Inversor Parameters' shows the following Python script:

```

1 # Carregar as bibliotecas DesignScript e padrão do Python
2 import sys
3 import clr
4 clr.AddReference('ProtoGeometry')
5 from Autodesk.DesignScript.Geometry import *
6
7 circ_panels = IN[0]
8
9 #Tratamento da lista de entrada para retirar os valores nulos e repetidos
10 circ_panels = [i for i in circ_panels if i and 'I' in i.Name]
11 inv = list(set(circ_panels))[0]
12 #Filtra parâmetros de instância e de tipo
13 all_parameters = list(inv.Parameters) + list(inv.Type.Parameters)
14 #Coloca os parâmetros em um dicionário
15 parameters_dict = {i.Name: i.Value for i in all_parameters}
16
17 OUT = parameters_dict
18
  
```

Fonte: o autor.

Figura 53 – Script completo e resultados para filtrar parâmetros desejados.



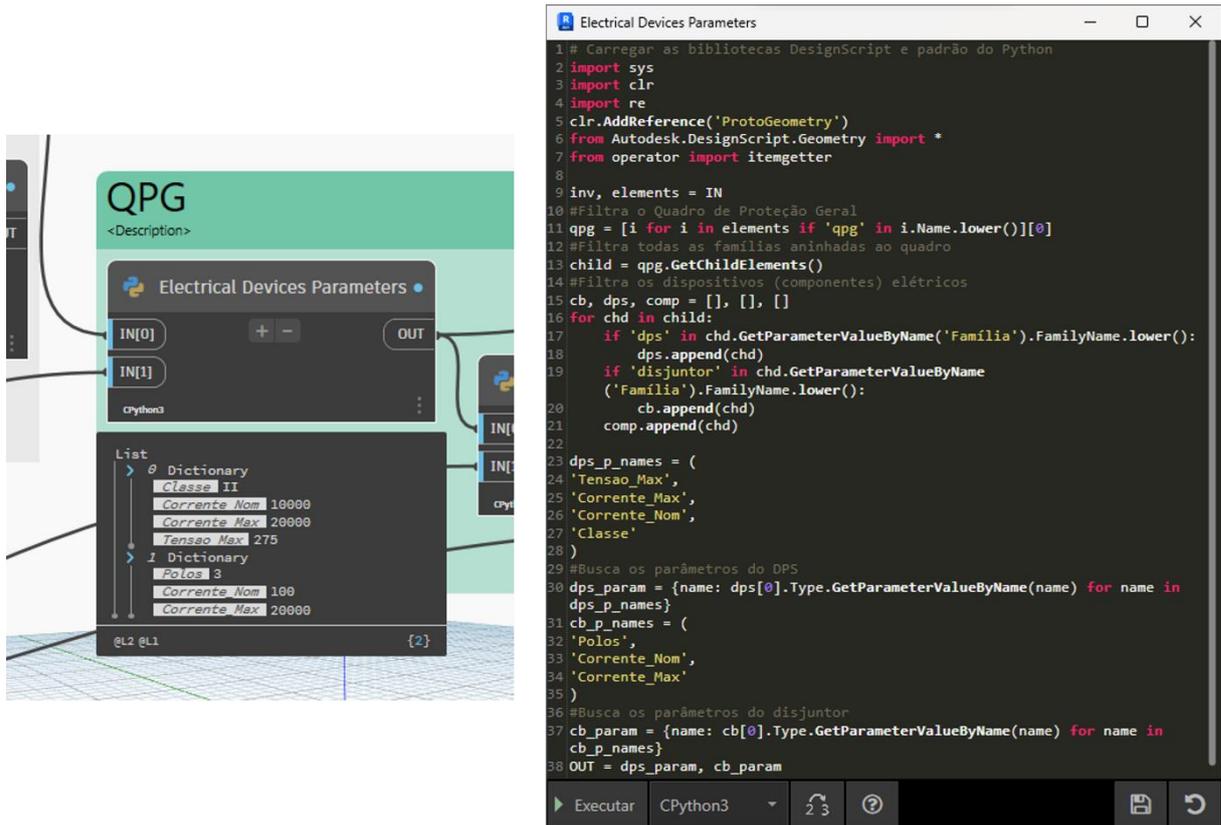
The image shows a Revit script execution interface. On the left, a 'Dictionary' window displays the output of the 'Filtered Parameters' script. The dictionary contains filtered parameters: 'Fases', 'Modelo', 'Strings/MPPT', 'Total MPPTs', 'Fabricante', and 'Potencia'. On the right, a code editor window titled 'Filtered Parameters' shows the following Python script:

```

1 # Carregar as bibliotecas DesignScript e padrão do Python
2 import sys
3 import clr
4 clr.AddReference('ProtoGeometry')
5 from Autodesk.DesignScript.Geometry import *
6
7 parameters = IN[0]
8
9 names = ('Potencia', 'Fabricante', 'Modelo', 'Fases', 'Total_MPPTs',
10 'Strings/MPPT')
11 OUT = {name: value for name,value in parameters.items() if name in names}
12
  
```

Fonte: o autor.

Figura 54 – Script completo e resultados para separação dos parâmetros dos dispositivos.

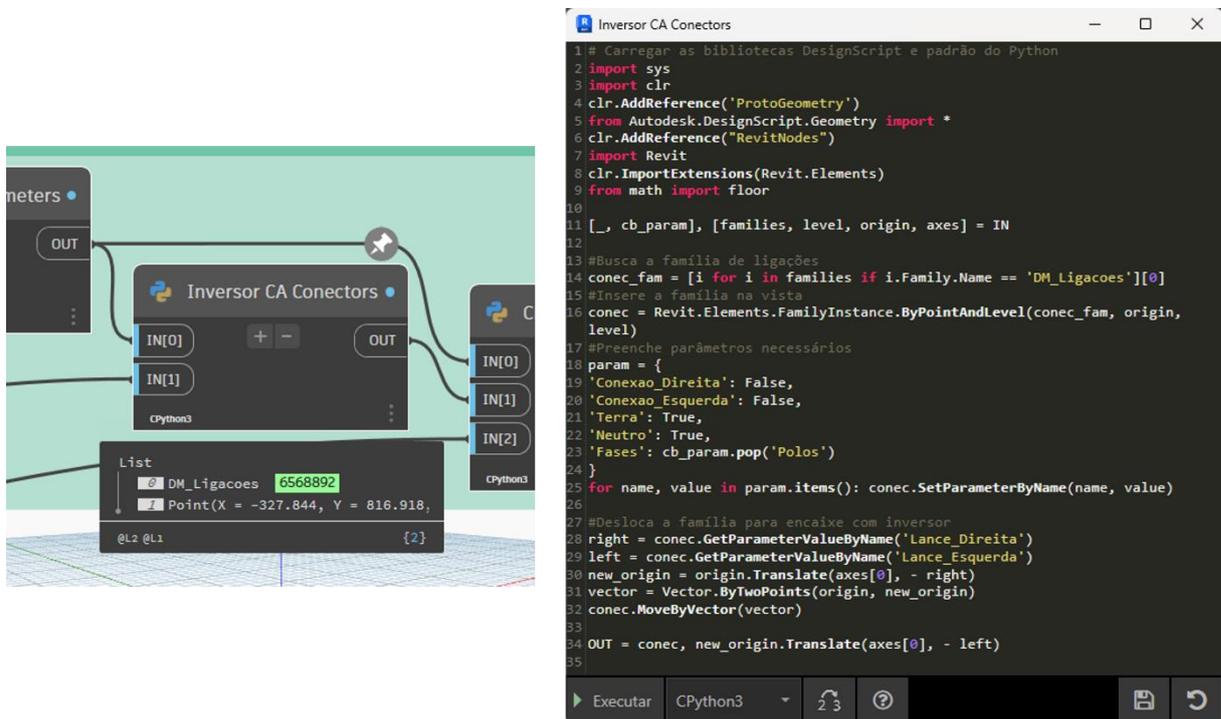


```

1 # Carregar as bibliotecas DesignScript e padrão do Python
2 import sys
3 import clr
4 import re
5 clr.AddReference('ProtoGeometry')
6 from Autodesk.DesignScript.Geometry import *
7 from operator import itemgetter
8
9 inv, elements = IN
10 #Filtra o Quadro de Proteção Geral
11 qpg = [i for i in elements if 'qpg' in i.Name.lower()][0]
12 #Filtra todas as famílias aninhadas ao quadro
13 child = qpg.GetChildElements()
14 #Filtra os dispositivos (componentes) elétricos
15 cb, dps, comp = [], [], []
16 for chd in child:
17     if 'dps' in chd.GetParameterValueByName('Familia').FamilyName.lower():
18         dps.append(chd)
19     if 'disjuntor' in chd.GetParameterValueByName
20         ('Familia').FamilyName.lower():
21         cb.append(chd)
22     comp.append(chd)
23
24 dps_p_names = (
25     'Tensao_Max',
26     'Corrente_Max',
27     'Corrente_Nom',
28     'Classe'
29 )
30 #Busca os parâmetros do DPS
31 dps_param = {name: dps[0].Type.GetParameterValueByName(name) for name in
32     dps_p_names}
33 cb_p_names = (
34     'Polos',
35     'Corrente_Nom',
36     'Corrente_Max'
37 )
38 #Busca os parâmetros do disjuntor
39 cb_param = {name: cb[0].Type.GetParameterValueByName(name) for name in
40     cb_p_names}
41 OUT = dps_param, cb_param
  
```

Fonte: o autor.

Figura 55 – Script completo e resultados para inserir instância de cabos entre inversor e QPG.

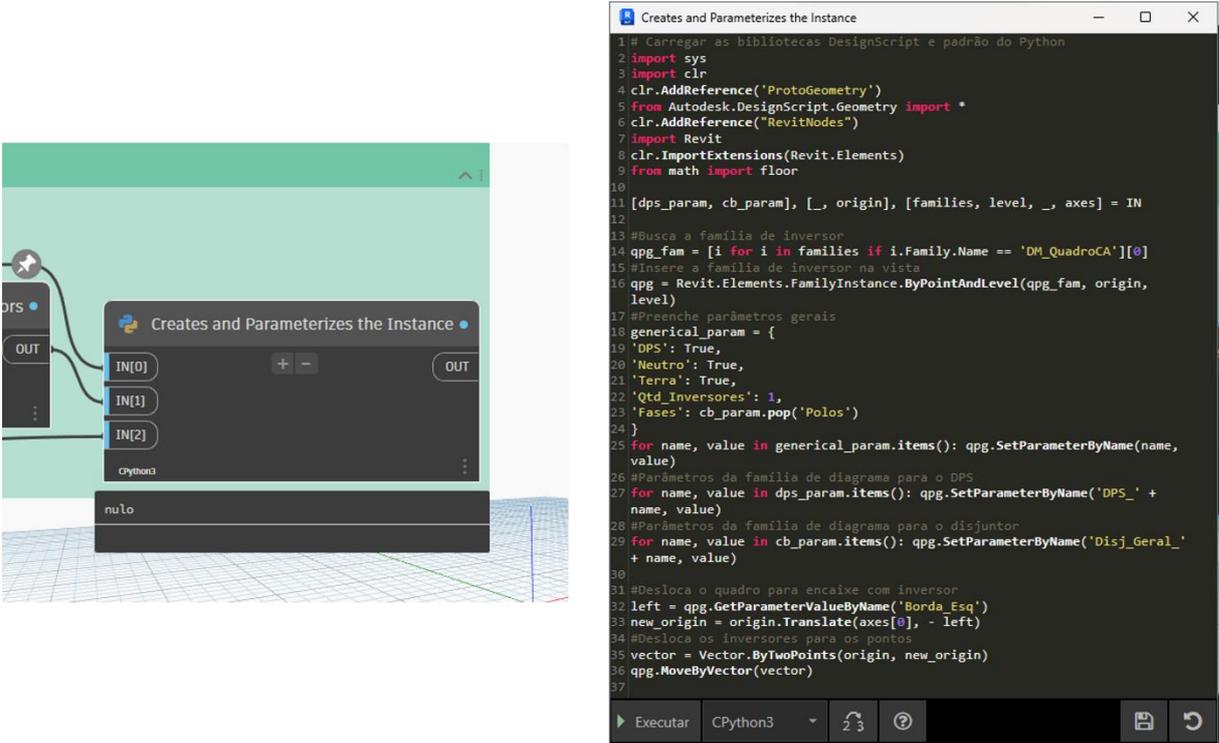


```

1 # Carregar as bibliotecas DesignScript e padrão do Python
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 from math import floor
10
11 [_, cb_param], [families, level, origin, axes] = IN
12
13 #Busca a família de ligações
14 conec_fam = [i for i in families if i.Family.Name == 'DM_Ligacoes'][0]
15 #Insere a família na vista
16 conec = Revit.Elements.FamilyInstance.ByPointAndLevel(conec_fam, origin,
17     level)
18 #Preenche parâmetros necessários
19 param = {
20     'Conexao_Direita': False,
21     'Conexao_Esquerda': False,
22     'Terra': True,
23     'Neutro': True,
24     'Fases': cb_param.pop('Polos')
25 }
26 #Desloca a família para encaixe com inversor
27 right = conec.GetParameterValueByName('Lance_Direita')
28 left = conec.GetParameterValueByName('Lance_Esquerda')
29 new_origin = origin.Translate(axes[0], - right)
30 vector = Vector.ByTwoPoints(origin, new_origin)
31 conec.MoveByVector(vector)
32
33 OUT = conec, new_origin.Translate(axes[0], - left)
  
```

Fonte: o autor.

Figura 56 – Script completo e resultados para inserir instância de quadro metálico.



The image shows a Dynamo script node titled "Creates and Parameterizes the Instance" with three input ports (IN[0], IN[1], IN[2]) and one output port (OUT). The output is labeled "nuLo". To the right, a Python script window displays the following code:

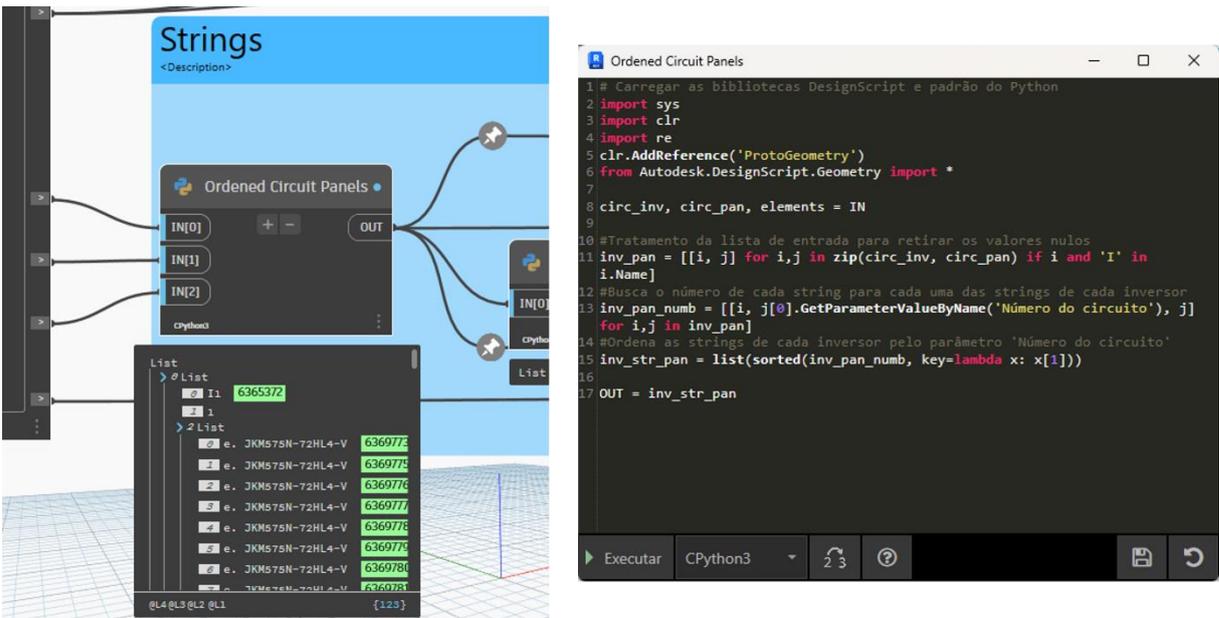
```

1 # Carregar as bibliotecas DesignScript e padrão do Python
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 from math import floor
10
11 [dps_param, cb_param], [_, origin], [families, level, __, axes] = IN
12
13 #Busca a família de inversor
14 qpg_fam = [i for i in families if i.Family.Name == 'DM_QuadroCA']
15 #Insere a família de inversor na vista
16 qpg = Revit.Elements.FamilyInstance.ByPointAndLevel(qpg_fam, origin,
17 level)
17 #Preenche parâmetros gerais
18 generical_param = {
19 'DPS': True,
20 'Neutro': True,
21 'Terra': True,
22 'Qtd_Inversores': 1,
23 'Fases': cb_param.pop('Polos')
24 }
25 for name, value in generical_param.items(): qpg.SetParameterByName(name,
26 value)
26 #Parâmetros da família de diagrama para o DPS
27 for name, value in dps_param.items(): qpg.SetParameterByName('DPS_' +
28 name, value)
28 #Parâmetros da família de diagrama para o disjuntor
29 for name, value in cb_param.items(): qpg.SetParameterByName('Disj_Geral_'
30 + name, value)
31
32 #Desloca o quadro para encaixe com inversor
33 left = qpg.GetParameterValueByName('Borda_Esq')
34 new_origin = origin.Translate(axes[0], - left)
35 #Desloca os inversores para os pontos
36 vector = Vector.ByTwoPoints(origin, new_origin)
37 qpg.MoveByVector(vector)
38

```

Fonte: o autor.

Figura 57 – Script completo e resultados para tratar as informações das strings.



The image shows a Dynamo script node titled "Ordered Circuit Panels" with three input ports (IN[0], IN[1], IN[2]) and one output port (OUT). The output is a list of strings. To the right, a Python script window displays the following code:

```

1 # Carregar as bibliotecas DesignScript e padrão do Python
2 import sys
3 import clr
4 import re
5 clr.AddReference('ProtoGeometry')
6 from Autodesk.DesignScript.Geometry import *
7
8 circ_inv, circ_pan, elements = IN
9
10 #Tratamento da lista de entrada para retirar os valores nulos
11 inv_pan = [[i, j] for i, j in zip(circ_inv, circ_pan) if i and 'I' in
12 i.Name]
13 #Busca o número de cada string para cada uma das strings de cada inversor
14 inv_pan_num = [[i, j[0].GetParameterValueByName('Número do circuito'), j]
15 for i, j in inv_pan]
16 #Ordena as strings de cada inversor pelo parâmetro 'Número do circuito'
17 inv_str_pan = list(sorted(inv_pan_num, key=lambda x: x[1]))
18
19 OUT = inv_str_pan

```

The output of the script is a list of strings, shown in the screenshot as:

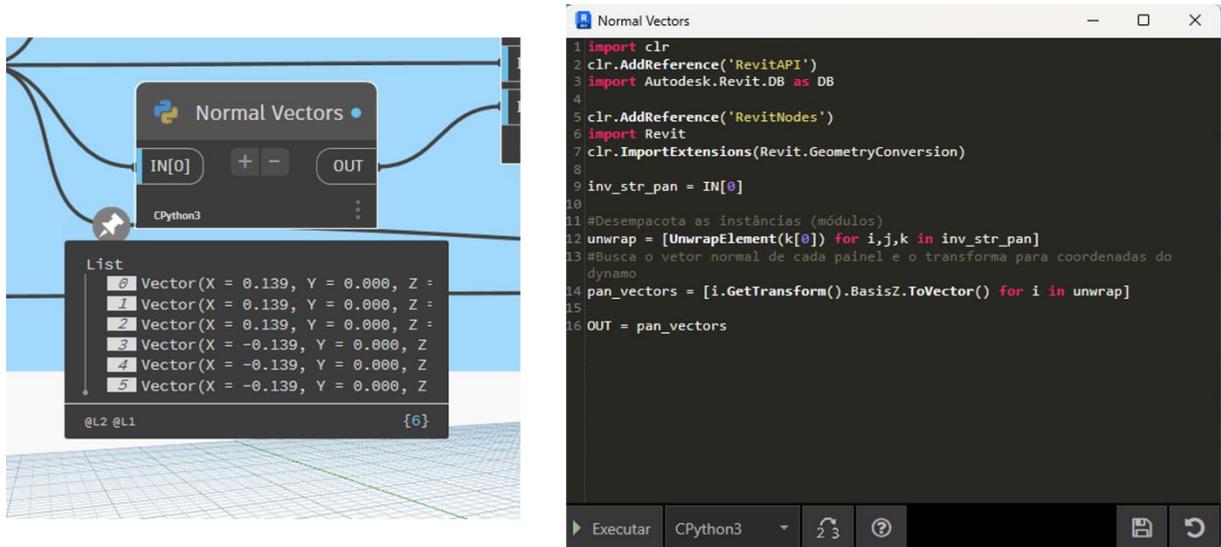
```

List
> List
  0 I 6365372
  1
  2 List
    0 e. JKM575N-72HL4-V 6369773
    1 e. JKM575N-72HL4-V 6369775
    2 e. JKM575N-72HL4-V 6369776
    3 e. JKM575N-72HL4-V 6369777
    4 e. JKM575N-72HL4-V 6369778
    5 e. JKM575N-72HL4-V 6369779
    6 e. JKM575N-72HL4-V 6369780
    7 e. JKM575N-72HL4-V 6369781

```

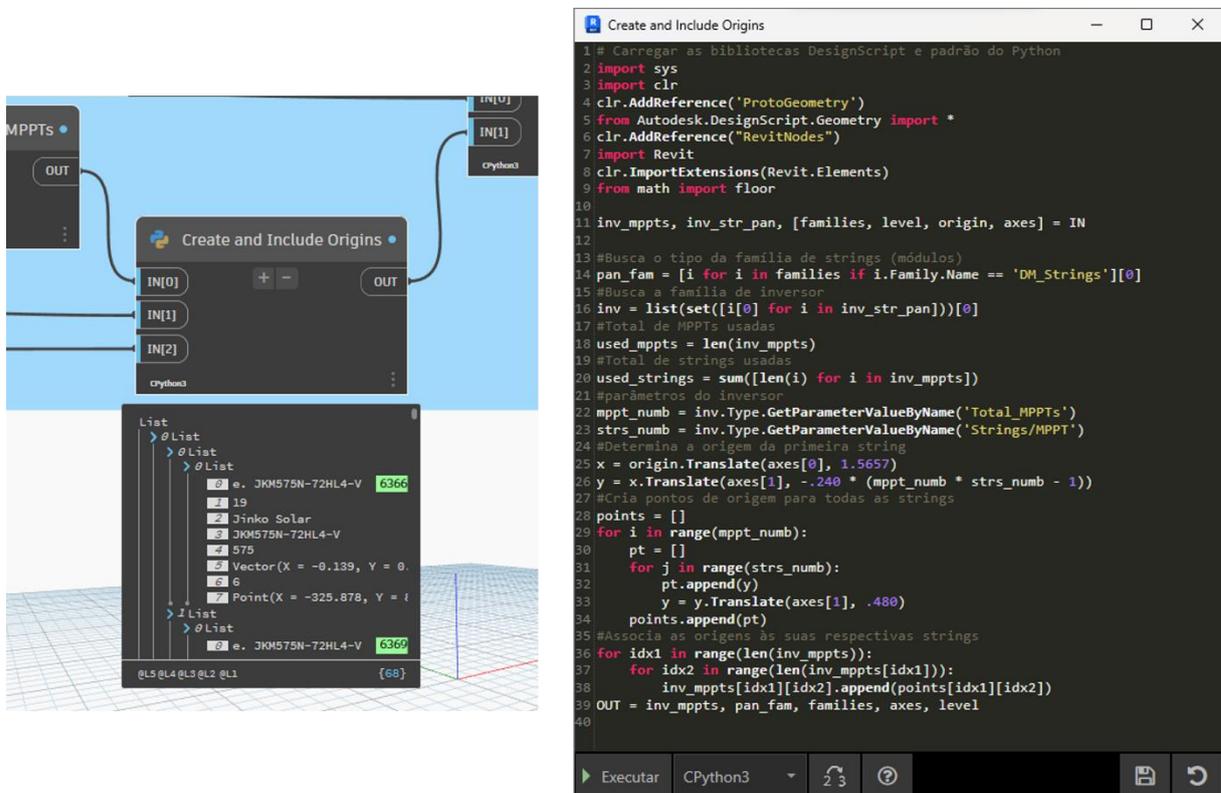
Fonte: o autor.

Figura 58 – Script completo e resultados para a busca de vetores normais aos módulos FV.



Fonte: o autor.

Figura 59 – Script completo e resultados para criar diagramas trifilares de strings em suas MPPTs.



Fonte: o autor.