



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE ARTES E COMUNICAÇÃO
DEPARTAMENTO DE ARQUITETURA E URBANISMO
CURSO DE GRADUAÇÃO EM ARQUITETURA E URBANISMO

EVELYN AYANE DE BRITO SOUZA

**EXPLORANDO AUTOMAÇÕES DE PROCESSOS DE MODELAGEM DE
PROJETO ARQUITETÔNICO EM BIM COM PROGRAMAÇÃO VISUAL**

Recife

2025

EVELYN AYANE DE BRITO SOUZA

**EXPLORANDO AUTOMAÇÕES DE PROCESSOS DE MODELAGEM DE
PROJETO ARQUITETÔNICO EM BIM COM PROGRAMAÇÃO VISUAL**

Trabalho de Conclusão de Curso apresentado ao Curso de Arquitetura e Urbanismo da Universidade Federal de Pernambuco – UFPE, como requisito parcial para obtenção do grau de Bacharel em Arquitetura e Urbanismo.

Orientador (a): Cristiana Maria Sobral Griz

Recife

2025

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

Souza, Evelyn Ayane de Brito.

Explorando automações de processos de modelagem de projeto arquitetônico em BIM com programação visual / Evelyn Ayane de Brito Souza. - Recife, 2025. 61 p. : il.

Orientador(a): Cristiana Maria Sobral Griz

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro de Artes e Comunicação, Arquitetura e Urbanismo - Bacharelado, 2025.

Inclui referências.

1. BIM. 2. Modelagem paramétrica. 3. Programação visual. I. Griz, Cristiana Maria Sobral. (Orientação). II. Título.

720 CDD (22.ed.)

DEDICATÓRIA

À minha mãe. Que sabe que a educação é o caminho. Que é professora da rede pública, e que hoje, forma mais uma filha no ensino superior. Mãezona.

AGRADECIMENTOS

Agradeço a Deus, por ser tão generoso e me permitir viver uma vida cheia de amor.

Aos meus pais, Edineide e Janildo, que me amam tanto e nunca me deixaram faltar nada. Que dão o melhor de si para que eu realize os meus sonhos. Obrigada mãe, obrigada pai. Eu amo vocês!

À minha irmã Hellen. Não imagino minha vida sem você como irmã mais velha. Obrigada por tanto amor e companheirismo. Juntas eu sei que podemos superar as adversidades da vida. Te amo. Rezo para que você seja imensamente feliz e realizada.

À minha tia Edilma, que me apoia todos os dias e fica tão feliz com a minha felicidade. Que vê tanto potencial em mim e sabe que eu amo escrever essas cartinhas de amor. Tia, obrigada por tudo. Me sinto tão amada e lhe amo o dobro de volta.

À minha avó. Que me ensinou na prática o que é um bom dengo. Que me levava pra escola desde quando eu era do tamanho de um tamborete. Eu dizia: “vovó, já passou o poste!!” e ela voltava o caminho comigo só pra me levar no colo, pois prometia que todo dia, do poste até em casa, eu iria no colo! Eu queria morar nessas memórias, vó. Te amo até a lua.

Ao meu gato Floquinho, que nunca vai ler esse agradecimento, mas que me ajudou, mesmo sem saber, nos anos da Pandemia e nas aulas remotas. Obrigada por ser danadinho, amoroso e puro do coração.

À minha orientadora Cris, que é referência quando escuto falarem sobre Modelagem Paramétrica. Obrigada pela orientação e por me acolher na célula BIM da UFPE, é um prazer poder aprender com vocês.

A todos os meus amigos que me apoiaram nesse caminho e que deixam a minha vida mais feliz. A Júlia, Camilly e Sabrina. Que bom poder contar com a amizade, a atenção e o carinho de vocês.

Ao GE 07, em especial Babi e Alice, que desde o primeiro período passávamos noites em claro para entregar o melhor trabalho que podíamos. Meninas, que bom ter conhecido vocês. Estamos conseguindo!

Por fim, a mim mesma. Por tanta dedicação e esforço nesses últimos anos, que mesmo começando com pouco e de uma cidade totalmente diferente, deu um jeito de entender e apreciar com sensibilidade as dinâmicas do novo cotidiano.

Gratidão.

RESUMO

Este trabalho investiga como a combinação entre modelagem paramétrica e programação visual pode automatizar tarefas repetitivas na fase de modelagem de projetos. Embora as tecnologias BIM tenham avançado significativamente, ainda existem desafios relacionados à execução de tarefas manuais e repetitivas na modelagem - problema central identificado nesta pesquisa. Diante desse contexto e dos desafios atuais da indústria AECO, como prazos curtos, projetos mais complexos e demanda por maior precisão, o estudo identifica, classifica e desenvolve automações em três categorias: criação de objetos BIM, documentação automática de projetos e verificação de normas técnicas. Utilizando o *Dynamo* para programação visual em ambiente BIM, as rotinas demonstram como parâmetros geométricos e não geométricos integram-se nessas automações. A metodologia adotou uma abordagem exploratória com desenvolvimento prático de rotinas, mostrando ganhos em produtividade e precisão. Como contribuição, apresenta um repertório de aplicações práticas adaptáveis a diferentes contextos projetuais.

Palavras-chave: BIM; modelagem paramétrica; programação visual.

ABSTRACT

This study investigates how the combination of parametric modeling and visual programming can automate repetitive tasks in project modeling phases. Although BIM technologies have advanced significantly, challenges remain regarding the execution of manual and repetitive modeling tasks – the core problem addressed in this research. Given this context and current AECO industry challenges (such as tight deadlines, increasingly complex projects, and demands for higher precision), the study identifies, classifies, and develops automations in three categories: BIM object creation, automated project documentation, and technical standards verification. Using Dynamo for visual programming in a BIM environment, the routines demonstrate how geometric and non-geometric parameters integrate into these automations. The methodology adopted an exploratory approach with practical routine development, demonstrating gains in productivity and precision. As a contribution, it provides a repertoire of practical applications adaptable to different project contexts.

Palavras-chave: BIM; parametric modeling; visual programming.

LISTA DE FIGURAS

Figura 1: Metadados de uma parede no <i>Autodesk Revit 2024</i>	22
Figura 2: Fluxo lógico da programação	25
Figura 3: Exemplo de programação textual em <i>JavaScript</i>	26
Figura 4: Exemplo de programação visual	27
Figura 5: Exemplo de programação visual feita no <i>software Grasshopper</i>	28
Figura 6: Exemplo de programação visual feita no <i>Dynamo</i>	29
Figura 7: Rotina do estudo	34
Figura 8: Reprodutor do <i>dynamo</i> para a rotina.....	35
Figura 9: Paredes do reservatório com armaduras introduzidas	36
Figura 10: Lógica para produção dos scripts	37
Figura 11: Rotina do estudo	38
Figura 12: Demonstração dos eixos criados	38
Figura 13: Planilha adotada no estudo 01	41
Figura 14: Família de pilar utilizada.....	41
Figura 15: Importação da planilha de pilar	42
Figura 16: Criação dos pilares a partir da planilha	43
Figura 17: Criação das paredes a partir dos pilares.....	45
Figura 18: Criação do piso	46
Figura 19: Rotina completa	47
Figura 20: Modelo explodido no <i>Revit</i>	47
Figura 21: Planilha adotada na rotina.....	49
Figura 22: Rotina do estudo	49
Figura 23: visualização dos níveis gerados na plataforma	50
Figura 24: Planilha adotada na rotina.....	51
Figura 25: Rotina no <i>dynamo</i>	52
Figura 26: Criação das pranchas na plataforma.....	52
Figura 27: Rotina do estudo	54
Figura 28: Projeto residencial adotado como estudo de caso para testagem da rotina	54
Figura 29: Exemplo de janela que não atendeu à especificação da norma	55

LISTA DE ABREVIATURAS E SIGLAS

BIM	BIULDING INFORMATION MODELING
IFC	INDUSTRY FOUNDATION CLASSES
MA	MODELAGEM ALGORÍTIMICA
MP	MODELAGEM PARAMÉTRICA
PC	PROGRAMAÇÃO COMPUTACIONAL
PV	PROGRAMAÇÃO VISUAL
UFPE	UNIVERSIDADE FEDERAL DE PERNAMBUCO

SUMÁRIO

INTRODUÇÃO	10
1 REVISÃO DA LITERATURA.....	18
1.1 Evolução das mídias de representação de projetos na arquitetura	18
1.2 <i>Building Information Modeling</i> (BIM)	20
1.3 Modelagem Paramétrica e Modelagem Algorítmica.....	23
2 A MODELAGEM PARAMÉTRICA COMO FERRAMENTA DE AUTOMATIZAÇÃO DO PROCESSO DE MODELAGEM DO PROJETO	31
2.1 Criação de Objetos BIM: Detalhamento de Paredes de Reservatórios de Concreto Armado usando a Ferramenta <i>Dynamo</i>	33
2.2 Documentação e pré-dimensionamento de projeto: Automatização no desenvolvimento de projetos sustentáveis de engenharia via metodologia BIM ...	36
3 EXPLORAÇÃO E DESENVOLVIMENTO DE AUTOMATIZAÇÕES RELACIONADAS À PROCESSOS DE MODELAGEM DE PROJETO, SOB A LÓGICA PARAMÉTRICA E ALGORÍTMICA	40
3.1 Criação de objetos BIM: pilares, paredes e pisos	40
3.1.1 Pilares	40
3.1.2 Paredes	43
3.1.3 Pisos.....	45
3.2 Documentação de Projeto.....	48
3.2.1 Criação de níveis automáticos	48
3.2.2 Geração de pranchas automáticas.....	50
3.3 Gestão de metadados: verificação da área de ventilação mínima de ambientes	53
CONSIDERAÇÕES FINAIS	57
REFERÊNCIAS BIBLIOGRÁFICAS	59

INTRODUÇÃO

Ao longo da história, a prática arquitetônica evoluiu significativamente em resposta às transformações tecnológicas. No Brasil, até as décadas de 1980 e 1990, a grande maioria dos projetos ainda era desenvolvida no lápis e papel, uma forma de representação que ainda hoje permite aos arquitetos explorarem conceitos e detalhes com sensibilidade e rigor técnico. Mas, embora essa prática atendesse às demandas de sua época, ela apresentava limitações em termos de tempo e adaptabilidade, pois alterações ou revisões de projeto exigiam muitos retrabalhos.

Paralelamente, a Indústria da Engenharia, Arquitetura, Construção e Operação (AECO) começou a pedir mais precisão, eficiência e agilidade. Com isso, vieram desafios relacionados ao tempo que se levava na execução dos projetos e a dificuldade de adaptá-los rápido às novas exigências. A transição dos projetos da arquitetura, que deixou de depender só do desenho à mão para integrar ferramentas tecnológicas, foi um processo gradual e influenciado pelas mudanças sociais, econômicas e tecnológicas que aconteciam na sociedade. Segundo Eastman *et al.* (2011, p. 23), essa evolução não só mostrou avanços técnicos, mas também uma necessidade de atender às demandas de uma nova indústria que estava chegando em constante transformação, caracterizada por prazos mais curtos, empreendimentos mais complexos e uma busca por soluções cada vez mais precisas e eficientes.

Um importante marco inicial dessa evolução aconteceu em 1982, com o lançamento do *AutoCAD* - o primeiro *software* amplamente utilizado para o desenho assistido por computador (DESOUZA, 2014). Com ele, os arquitetos passaram a contar com uma ferramenta que potencializava o processo de representação do projeto com mais precisão e agilidade na elaboração dos desenhos. É notável que o *software* simplificou a execução de ajustes e variações de projeto. No entanto, apesar de suas contribuições, sua abordagem baseada em gráficos vetoriais¹ apresentava limitações importantes. Por exemplo, no *AutoCAD*, a representação de uma parede é apenas um conjunto de linhas, não tem informações agregadas a ela, como seu material, custo ou propriedades térmicas, informações que são preciso saber para a execução de uma obra. É possível observar que essa limitação de ferramentas como o *Autocad* impacta diretamente a integração de diferentes disciplinas de projeto

¹ Mais bem explicado no Capítulo 01.

(arquitetura, estrutura, instalações elétricas e hidrossanitárias), pois qualquer modificação que tenha, vai exigir atualizações manuais em todos os desenhos afetados. Em projetos complexos, como é o caso dos da AECO, isso faz com que os profissionais tenham que gerenciar centenas de arquivos separados, dificultando não só a sincronização entre plantas, cortes e elevações, mas também a coordenação entre as disciplinas. O resultado é um processo muito trabalhoso e passível de erros, porque cada alteração, como a movimentação de uma simples parede, exige atualizações manuais em todos os documentos e a verificação cruzada com outras disciplinas para checar se não houve nenhuma interferência. Essa fragmentação é um grande desafio para atender às demandas de eficiência, precisão entre diferentes áreas e colaboração em tempo real.

A partir da década de 2010, consolidou-se a chamada Indústria 4.0 - quarta revolução industrial marcada pela integração de tecnologias digitais, como Internet das Coisas, Inteligência Artificial e Computação em nuvem (KAGERMANN *et al.*, 2013). Na AECO, esse movimento trouxe uma necessidade de sistemas que permitissem o trabalho colaborativo entre todos os agentes do processo construtivo - desde arquitetos e engenheiros no escritório até as equipes no canteiro de obras, e era necessário superar a fragmentação de informações do CAD.

Neste contexto, o *Building Information Modeling* (BIM), ou Modelagem da Informação da Construção, surge como resposta. De acordo com Eastman *et al.* (2011, p. 5), o BIM pode ser definido como "uma abordagem integrada baseada em modelos digitais ricos em dados que suportam a colaboração entre todos os envolvidos no planejamento, design, construção e operação de uma edificação ou infraestrutura". Essa definição representa uma dimensão tecnológica de um ecossistema ainda mais amplo. Como destacam Sacks *et al.* (2018), o BIM também é um processo de gestão da informação construída, que demanda mudanças organizacionais na forma como os *stakeholders* – todos os agentes envolvidos nesse processo, como arquitetos, engenheiros, construtores, clientes e gestores públicos – compartilham e atualizam dados ao longo do ciclo de vida do projeto.

Essa visão é reforçada por Succar (2009), que caracteriza o BIM como uma metodologia com três pilares que se completam: o tecnológico, que engloba ferramentas digitais e modelos paramétricos; o processual, referente aos fluxos colaborativos entre agentes; e o estratégico, que engloba políticas de padronização e governança de dados. Essa definição mostra que o BIM não é só uma ferramenta

digital, mas um novo jeito de pensar e projetar que transforma as práticas de projeto e construção ao conectar tecnologia, processos e pessoas. O grande avanço e evolução, então, não está nos computadores, mas sim na forma de colaboração e gerenciamento que o BIM propõe.

Em relação ao ramo tecnológico do BIM, o que define uma plataforma como sendo BIM (como o *Revit*, *Archicad*, *Eberick*, etc.) é ter a modelagem paramétrica (MP)² e a interoperabilidade³ como seus principais pilares. Por ser paramétrico, as alterações em uma parte do modelo refletem automaticamente em todas as outras vistas e elementos vinculados. Como as documentações⁴ do projeto – plantas, cortes e fachadas, estão todas associadas ao modelo, qualquer modificação no modelo também é sincronizada em tais documentos em tempo real.

Além disso, essas plataformas centralizam informações de natureza não geométrica – como tipos de materiais, desempenho térmico e custos, denominadas metadados. Tais informações, que antes ficavam espalhadas nos diversos arquivos de diferentes disciplinas, são vinculadas a cada elemento construtivo, que transformam esse modelo num repositório de conhecimento técnico. Os metadados representam um grande avanço do BIM no âmbito tecnológico. Enquanto no CAD uma parede é só um conjunto de linhas, no BIM ela pode carregar consigo dados como resistência térmica, custo por metro quadrado, compatibilidade com sistemas estruturais, entre outros. Como explica Sacks *et al.* (2018, p. 112),

"Os metadados no BIM permitem que elementos carreguem não apenas geometria, mas significado semântico, transformando o modelo num banco de dados relacional onde paredes 'conhecem' seus materiais, janelas 'calculam' seu desempenho térmico, e instalações 'registram' seus prazos de manutenção".

A MP é um dos temas abordados no design computacional⁵. Nesse tipo de abordagem, os elementos do modelo são controlados por regras parametrizadas, e

² Mais bem definida no Capítulo 01.

³ Este termo será explicado no Capítulo 01.

⁴ A documentação técnica é um dos diversos usos do BIM. Como destacam Eastman *et al.* (2011, p. 87), os usos BIM são aplicações práticas da modelagem para resolver desafios em fases do projeto (desde simulações iniciais até a gestão da obra), sempre com objetivo de aumentar a precisão, otimizar tempo ou viabilizar análises complexas. Então, documentar (um de seus usos) consiste em extrair automaticamente plantas, cortes e tabelas diretamente do modelo, garantindo que todas as representações técnicas estejam sempre sincronizadas.

⁵ Segundo Oxman (2014), este pode ser entendido como uma área de pesquisa dedicada ao desenvolvimento de métodos e ferramentas que utilizam tecnologias computacionais para facilitar a criação de projetos.

quando um parâmetro é alterado, todos os elementos vinculados a ele se ajustam automaticamente. Quando aplicada em ambientes BIM, como destacam Griz, Andrade e Resende (2021), a MP amplia seu potencial: parâmetros geométricos passam a integrar os metadados, criando, também, parâmetros não geométricos, como custos, desempenho térmico e funções construtivas, etc. É aqui que se encontra a disrupção no processo projetual: a partir desses parâmetros não-geométricos é possível gerar ou modificar geometrias automaticamente a partir dessas informações, permitindo assim, simulações (testes para medir o desempenho de uma solução), análises e otimizações que podem ser executadas diretamente no modelo integrado.

É importante frisar que, para que esse potencial seja plenamente alcançado, é preciso que o modelo seja estruturado com informações bem definidas e organizadas. Não é só criar geometrias em 3D. É preciso estabelecer o nível de informação para atender à finalidade que o modelo está sendo desenvolvido - quais os dados cada elemento vai carregar, como eles serão classificados e de que forma se relacionam com as outras disciplinas. A qualidade das análises e otimizações depende do planejamento do modelo, onde se decide não apenas como modelar, mas o que o modelo deve representar em termos de informações.

Nesse contexto, a modelagem algorítmica⁶ (MA) vem para potencializar o processo paramétrico. Enquanto *softwares* de MP operam através da manipulação direta na geometria do modelo - como é o caso da parametrização de famílias⁷ no *Revit*, a MA avança ao permitir criar as geometrias através de *scripts*⁸. Como explica Leach (2014, p. 34), com a MA é possível ir além das limitações das ferramentas padrão, já que o projetista passa a manipular não a geometria em si, mas as instruções computacionais que a geram - um salto que amplia as possibilidades de personalização e controle sobre o modelo.

A MA pode ser textual ou visual. A programação textual consiste na criação de algoritmos⁹ por meio de códigos escritos em linguagens baseadas em texto, enquanto a programação visual (PV) acontece em *softwares* com interfaces gráficas que

⁶ Método de criação de modelos digitais que utiliza algoritmos (mais bem explicado no Capítulo 01) para gerar geometrias e comportamentos em *softwares* de projeto.

⁷ As famílias são bibliotecas de objetos BIM parametrizados que compartilham algumas propriedades imutáveis, mas permitem aos usuários controlar dimensões, proporções e outras características geométricas de forma intuitiva.

⁸ *Scripts* são códigos de programação escritos para automatizar tarefas ou criar regras específicas em *softwares* (AUTODESK, 2021).

⁹ Definidos no Capítulo 01.

substituem o código textual por elementos visuais interconectados (nós, blocos ou fluxogramas), permitindo criar algoritmos através de manipulação direta na tela. Essa abordagem proporciona uma compreensão muito mais intuitiva do processo algorítmico, ao organizar as informações e as relações entre os componentes de uma forma espacial e visual (CYPHER *et al.*, 2010; GREEN, 2021).

A grosso modo, a MP, quando potencializada pela PV, pode atuar em dois campos complementares: na concepção projetual, permitindo que a ferramenta seja um agente ativo no processo de criação da forma, possibilitando a exploração de soluções criativas com ajustes dinâmicos; e na automatização de processos, otimizando tarefas repetitivas de modelagem. Esta pesquisa concentra-se no segundo aspecto, demonstrando como a PV pode melhorar fluxos de trabalho, liberando os profissionais para atividades de maior valor intelectual. Essa necessidade se justifica uma vez que as ferramentas BIM se destacam por sua capacidade de organizar e gerenciar informações complexas de forma centralizada. Porém, apesar desses avanços, muitos profissionais continuam tendo um processo de modelagem que envolve muitas tarefas repetitivas (e não intelectuais ou técnicas), limitando o alcance de seus benefícios que essas tecnologias poderiam trazer.

Para obter os melhores resultados na modelagem BIM, é essencial que os profissionais tenham métodos que aumentem a produtividade, economizem tempo em tarefas manuais repetitivas e assegurem a consistência de seus modelos. Isso é possível com o uso da programação visual, que permite automatizar processos de modelagem. Com a economia de tempo que o uso dessa tecnologia traz, os profissionais da AECO poderiam investir em atividades mais relevantes para o desenvolvimento do projeto, como a criatividade, o aperfeiçoamento de ideias e o próprio processo projetual em si. Dessa forma, deixariam de ser sobrecarregados por tarefas repetitivas e manuais, comuns no dia a dia no processo de modelagem digital, e passam a se concentrar no que realmente importa para a qualidade e inovação dos projetos.

Em síntese, apesar dos avanços trazidos pela tecnologia BIM, sua prática ainda enfrenta desafios, principalmente em relação à execução de tarefas repetitivas e manuais relacionadas à modelagem de projetos. Sendo assim, o objetivo geral desta pesquisa consiste em explorar como o uso da modelagem paramétrica e da programação visual, enquanto ferramenta BIM, pode automatizar o processo de

modelagem BIM, destacando os impactos dessa automação na eficiência, qualidade e execução de projetos no setor da Arquitetura e Urbanismo.

Para tanto, é preciso:

- Compreender a modelagem paramétrica e a programação visual e seu papel dentro do processo de modelagem BIM.
- Identificar e classificar tarefas relacionadas à modelagem de projetos que podem ser automatizadas.
- Desenvolver rotinas de automação de acordo com as classificações identificadas, demonstrando seu funcionamento e aplicabilidade prática.

Para alcançar estes objetivos, esta pesquisa adotou uma abordagem exploratória, fazendo revisão bibliográfica com desenvolvimento prático a fim de investigar como a MP e a PV podem ser integradas e ajudar nas fases de processo de modelagem de projeto. Sendo assim, esta pesquisa foi desenvolvida em três capítulos.

O Capítulo 1 apresenta a revisão da literatura, partindo da evolução das formas de representação gráfica como ferramenta fundamental para a expressão técnica e espacial em arquitetura - papel que hoje é aprimorado pelas tecnologias digitais. O estudo percorre essas ferramentas digitais, desde os sistemas CAD até a consolidação do BIM, com ênfase nos conceitos-chave como a interoperabilidade e a MP. Em seguida, avança para a discussão sobre a MA, que, como visto, divide-se em abordagens textuais e visuais - sendo esta última a ser explorada no trabalho. Para fundamentar esse estudo, o capítulo sintetiza um levantamento de referências nacionais e internacionais sobre BIM, MP, MA e PV, abrangendo artigos acadêmicos, teses, dissertações, manuais dos *softwares*, relatórios técnicos e etc. Além disso, explora como a Indústria 4.0 vem reconfigurando a prática arquitetônica através da junção entre modelos digitais e processos colaborativos.

O Capítulo 2 investiga a modelagem paramétrica como ferramenta de automatização do processo de modelagem de projetos arquitetônicos, com foco nas potencialidades da PV. O estudo identifica e classifica tarefas recorrentes em

modelagem feitas em ambiente BIM que podem ser otimizadas por meio de rotinas¹⁰ parametrizadas, organizando-as em três categorias a fim de exploração: criação de objetos BIM, documentação de projeto e verificação automática de normas técnicas. Para cada uma dessas categorias, analisa-se como a automação pode não apenas acelerar processos, mas também introduzir mais precisão e controle. Essa investigação é complementada por dois estudos de caso que demonstram ganhos em eficiência e qualidade em seus respectivos contextos. Por fim, reforça-se que a integração entre MP e PV pode proporcionar ganhos significativos em produtividade, tempo e eficiência nos processos de modelagem de projetos.

O Capítulo 3 traz a exploração prática das automações para processos de modelagem de projetos, seguindo a lógica paramétrica e algorítmica. Partindo das tarefas recorrentes identificadas e classificadas no Capítulo 2 - selecionadas tanto por sua aplicabilidade prática (com potencial de replicação em outros contextos) quanto pela viabilidade dentro do cronograma da pesquisa, o estudo desenvolve e demonstra as rotinas de automação. Utilizando o *Autodesk Revit 2024* como plataforma BIM, combinada com o *Dynamo* para PV, as rotinas exploram a lógica paramétrica por trás desses processos e o passo a passo algorítmico delas. Essa abordagem metodológica foi escolhida por oferecer um equilíbrio entre acessibilidade (para profissionais sem formação em programação textual) e eficiência, aliada à familiaridade da autora com as ferramentas. Os resultados validam não apenas os ganhos operacionais previstos, mas também demonstram como a integração entre MP e PV têm potencial para melhorar o processo de modelagem de projetos arquitetônicos.

As conclusões desta pesquisa reforçam o potencial da MP e PV na prática profissional. Os resultados demonstram que, quando integradas em ambientes BIM - desde a modelagem até a verificação técnica, essas abordagens estabelecem um fluxo de trabalho mais eficiente e produtivo. Além disso, a partir do entendimento da MP, é possível transcender a fase de modelagem de projeto e auxiliar também a fase de concepção, tornando-a estratégica para a otimização de soluções e tomada de decisão. Como contribuição, o trabalho consolida um repertório de aplicações adaptáveis a diversos contextos na fase de modelagem de projetos, servindo tanto

¹⁰ O termo "rotinas" refere-se a sequências lógicas de comandos programados que automatizam tarefas repetitivas. Na literatura, os termos "scripts", "algoritmos" ou "rotinas" são usados como sinônimos, porém neste trabalho optou-se por utilizar o termo "rotinas" para manter a padronização terminológica.

como referência técnica quanto estímulo para a evolução desses métodos na indústria AECO.

1 REVISÃO DA LITERATURA

A arquitetura tem historicamente utilizado a representação gráfica como ferramenta para dar forma às ideias e planejar espaços. Desde os primeiros registros gráficos encontrados na Mesopotâmia, até os sistemas digitais contemporâneos, a evolução das ferramentas projetuais mostra uma busca por precisão técnica e ampliação das possibilidades criativas (GIEDION, 2008). Nessa linha de raciocínio, Fraser e Henmi (2013) falam que a representação gráfica vai além do que a função só instrumental, ela é também um espaço para experimentar e pensar espacialmente. Para esses autores, o domínio dessas linguagens de representação - que agora são potencializadas por recursos digitais, auxiliam ainda mais o processo de projeto desde os primeiros esboços até a obra construída. Assim, a evolução das ferramentas de representação, descrita resumidamente a seguir, está intrinsecamente ligada ao aprimoramento do pensamento projetual.

1.1 Evolução das mídias de representação de projetos na arquitetura

Nos primórdios da arquitetura, o conhecimento construtivo era transmitido de forma prática, por meio da experiência manual passada de geração em geração. Como explica Silva (2008), esse saber empírico (conhecimento adquirido através de experiência e prática), embora valioso, tinha limitações por não ser registrado de uma forma mais sistemática. Então, as primeiras tentativas de documentação surgiram em resposta a essas necessidades práticas, como os símbolos cuneiformes da Mesopotâmia. Esses registros, mesmo simples, marcaram o início da conversão do conhecimento construtivo em representações gráficas, que mais tarde se tornaram indispensáveis no processo projetual arquitetônico.

O avanço das representações gráficas foi acelerado com a introdução de esquemas técnicos e desenhos em suportes permanentes, especialmente durante a Idade Média e o Renascimento. Como destacam Carpo (2017) e Pérez-Gómez (2016), o desenho arquitetônico consolidou-se não só como linguagem de comunicação, mas também como um instrumento cognitivo - permitindo tanto o planejamento quanto a antecipação de problemas construtivos. Essa evolução se acelerou ainda mais com a Revolução Industrial, pois, junto com ela veio o uso de tecnologias de impressão e a incorporação de máquinas ao trabalho, que facilitaram

a produção em grande escala e a padronização dos processos, impactando o campo da AECO.

Essa busca por mais eficiência e precisão continuou mudando a prática arquitetônica nos séculos seguintes. Desde a mecanização na Primeira Revolução Industrial (final do século XVIII, com a introdução de máquinas a vapor e produção em série), passando pela eletrificação e produção em massa da Segunda Revolução (final do século XIX, com linhas de montagem e energia elétrica), até os avanços digitais da Terceira, com automação e computação, a arquitetura acompanhou os ciclos de evolução tecnológica, adotando novos materiais, como o aço e o concreto armado nas edificações, e tendo processos mais rápidos e sistemáticos. Essas mudanças aumentaram as possibilidades criativas e produtivas dos projetistas, mas para isso, era preciso, também, que as ferramentas de representação dessas ideias evoluíssem junto.

Com a necessidade de lidar com empreendimentos mais complexos e atender às demandas por agilidade e precisão, a prática de representação de projeto incorporou ferramentas como o *Computer-Aided Design* (CAD). Desenvolvido no final dos anos 1960 e amplamente difundido nas décadas seguintes, o CAD trouxe ferramentas digitais para a elaboração de desenhos técnicos, baseados em elementos vetoriais como linhas, curvas e formas. Kalay (2004) destaca que, além de digitalizar o processo de desenho, o CAD permitiu a exploração da modelagem de volumetrias fixas (objetos 3D com formas definidas que não são dinâmicas ou paramétricas), contribuindo na representação espacial no âmbito digital.

Com o avanço da tecnologia, os sistemas CAD evoluíram junto. *Softwares* como *Autodesk AutoCAD* e *BricsCAD* passaram a oferecer ferramentas mais sofisticadas, incluindo a criação de sólidos, superfícies e malhas 3D. Eastman *et al.* (2011) destacam que o *AutoCAD* foi um marco nessa evolução ao trazer recursos como camadas, escalas automáticas e ferramentas de edição avançada. Essas funcionalidades não só agilizaram o trabalho dos arquitetos, como também elevaram a qualidade dos projetos, garantindo mais precisão gráfica e melhorando o processo de modelagem de projeto.

A evolução dos sistemas CAD, mesmo tendo ajudado muito os projetistas que se apropriaram das ferramentas, foi se mostrando insuficiente com o tempo. Enquanto a arquitetura demandava projetos cada vez mais complexos e integrados com outras disciplinas projetuais, essas ferramentas ficaram presas nas representações vetoriais,

que, mesmo eficientes para representação geométrica, os elementos arquitetônicos nelas são apenas abstrações lineares. Por exemplo, uma janela é apenas um retângulo composto por quatro segmentos, sem qualquer informação sobre seu material, desempenho acústico, etc. Como mostra Kymmell (2008), essa “cegueira informacional” faz com que os projetistas tenham que gerenciar centenas de dados (orçamentos, especificações, cronogramas, etc.) em várias planilhas e documentos externos - um processo que naturalmente é mais propenso a erros.

O desafio também vai além da falta de atributos não geométricos (como a falta de especificações de materiais, custos e outras informações): a arquitetura contemporânea exige coordenação em tempo real entre as disciplinas, já que enquanto o arquiteto define os espaços, o engenheiro estrutural calcula cargas e o projetista elétrico planeja circuitos e assim em diante. No entanto, nos sistemas CAD tradicionais, essa colaboração pode ser prejudicada pela fragmentação dos documentos, onde plantas, cortes e detalhes não estão vinculados entre si (são arquivos independentes que não se atualizam automaticamente). Um exemplo disso é que qualquer ajuste na dimensão de uma parede exige revisões manuais em vários documentos, assim como a verificação individual e cruzada com cada especialista.

Sobre essa fragmentação dos documentos relativos a um projeto da AECO, Eastman *et al.* (2011, p. 64) destacam que, em projetos de média complexidade, a falta de integração nesses fluxos de trabalho em CAD pede um "significativo esforço manual para reconciliar inconsistências entre disciplinas", como ocorre quando alterações em plantas arquitetônicas exigem horas de redesenho nas plantas estruturais e de instalações. Essa realidade mostra a necessidade de abordagens mais integradas e colaborativas, como acontece com o BIM (comentado a seguir).

1.2 *Building Information Modeling* (BIM)

No final do século XX, a crescente complexidade dos empreendimentos na AECO impulsionou a necessidade de ferramentas mais integradas e dinâmicas, o que culminou no surgimento do *Building Information Modeling* (BIM). Esse conceito começou na década de 1970, com o *Building Description System* (BDS), por Charles Eastman, que sugeriu uma solução inovadora para centralizar e integrar informações em um único modelo digital. Segundo Eastman *et al.* (1974), o BDS demonstrava que um sistema computacional poderia replicar os benefícios do desenho tradicional, ao

mesmo tempo em que superasse a fragmentação de informações e a falta de comunicação entre disciplinas.

A evolução dessa ideia levou à formalização do termo BIM nos anos 1990, consolidando-o como uma metodologia que combina tecnologias, políticas e processos para criar, utilizar e gerenciar modelos digitais ao longo de todo o ciclo de vida de um artefato. O Decreto nº 10.306, de 2 de abril de 2020, reforça essa definição ao descrever o BIM como:

“um conjunto de tecnologias e processos integrados que permite a utilização e atualização de modelos digitais de uma construção, de modo colaborativo, que sirva a todos os participantes do empreendimento, em qualquer etapa do ciclo de vida da construção”.

Essa definição destaca a capacidade do BIM não só no campo tecnológico, mas também reforça o caráter colaborativo que permite a interoperabilidade entre as diferentes disciplinas e etapas do projeto.

A interoperabilidade, um dos conceitos centrais no BIM, refere-se à capacidade de sistemas distintos compartilharem e interpretar dados coerentemente através do *Industry Foundation Classes* (IFC) - padrão aberto¹¹ que permite a troca de informações geométricas e não geométricas entre diferentes plataformas BIM, independentemente do *software* utilizado (*buildingSMART International*, 2020). Por exemplo, um arquiteto pode desenvolver seu projeto em uma determinada plataforma BIM, enquanto um engenheiro estrutural trabalha em outra, e ambos os modelos podem ser integrados em um modelo federado (modelo único que integra as contribuições das diversas disciplinas) através de um Ambiente Comum de Dados¹² (CDE). Como explica Succar (2009), esse processo cria um banco de dados dinâmico que serve como base para decisões mais estratégicas no projeto, além de tornar mais viável o processo de compatibilização¹³.

As tecnologias BIM adotam essa abordagem centrada na informação, transformando cada elemento do modelo digital em um repositório de metadados.

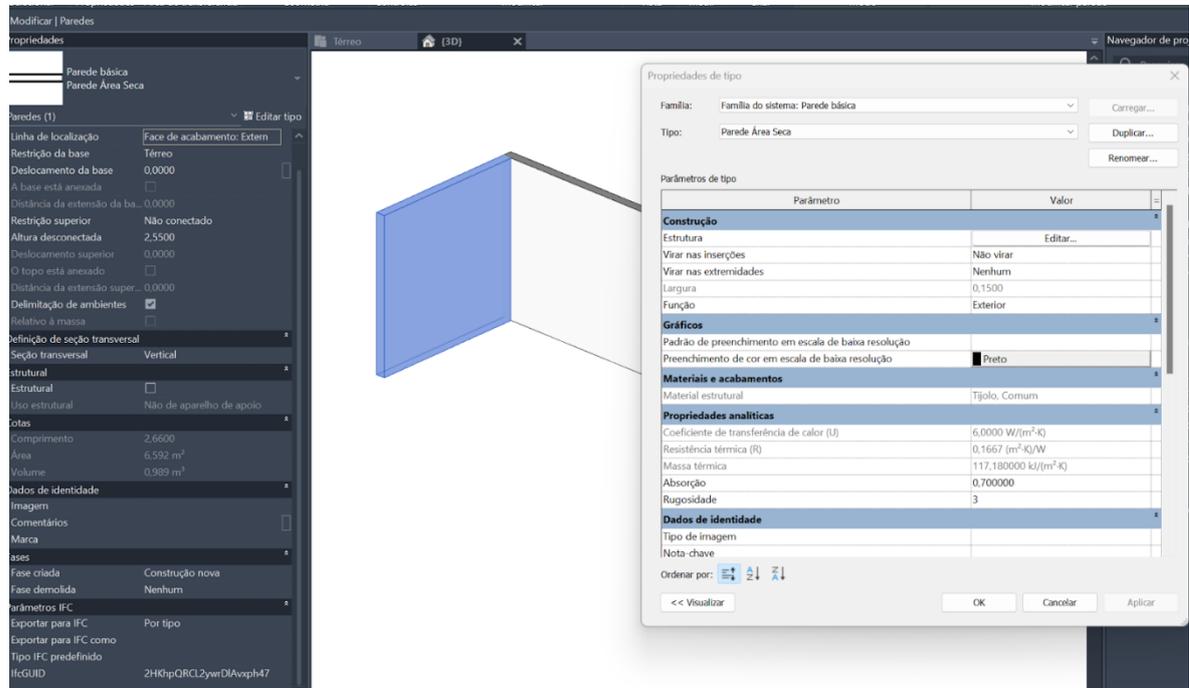
¹¹ Formatos ou protocolos técnicos públicos, documentados e livres de restrições de uso, que garantem interoperabilidade entre sistemas diferentes sem depender de um único fornecedor comercial.

¹² CDE (*Common Data Environment*) é um ambiente digital padronizado para armazenamento e compartilhamento de informações de projetos BIM.

¹³ Processo de garantir que todos os projetos envolvidos numa obra sejam compatíveis entre si, evitando interferências e conflitos entre as disciplinas.

Como comentado na Introdução, uma parede feita em ambiente BIM pode carregar consigo vários dados, como mostra a Figura 1.

Figura 1 - Metadados de uma parede no Autodesk Revit 2024.



Fonte: Acervo pessoal da autora

Smith (2020) explica que esses metadados, para serem compartilhados, seguem padrões abertos (como o já mencionado IFC), permitindo que sejam interpretados e atualizados por diferentes plataformas BIM ao longo do ciclo de vida do projeto (da concepção à demolição). É assim que é possível viabilizar ambientes colaborativos, onde arquitetos, engenheiros e construtores podem acessar, validar e atualizar informações em tempo real, independentemente das plataformas utilizadas em cada disciplina.

Percebe-se, então, que o BIM vai muito além de apenas modelar um edifício em 3D. É também sobre a forma como as informações sobre um empreendimento são gerenciadas e sobre uma representação digital que é capaz de seguir fielmente a construção real do edifício. Essa mesma representação pode armazenar desde detalhes técnicos até informações sobre materiais e prazos. Mas para que isso funcione efetivamente, não basta apenas modelar em uma plataforma BIM. É preciso mudar a forma como as equipes trabalham e como os dados são compartilhados,

definir previamente e estruturar o modelo, para assim ter um sucesso maior na sua execução. Ingram (2020) destaca que esse potencial só se realiza de fato quando as ferramentas digitais são combinadas com novos métodos de trabalho e uma cultura de compartilhamento contínuo de dados. Essa integração permite que todas as etapas – do projeto à operação do edifício, se beneficiem de informações centralizadas e sempre atualizadas, então aí sim seria o grande diferencial metodológico do BIM.

1.3 Modelagem Paramétrica e Modelagem Algorítmica

No contexto da documentação de projetos, a MP representa um potencial avanço na forma como se desenvolvem as geometrias da arquitetura e engenharia. Ela pode ser entendida como a "área da computação gráfica que representa a forma e as propriedades geométricas do modelo e que se utiliza de diferentes disciplinas da matemática para essa função, principalmente, a geometria." (POLONINI, 2014). Segundo a autora, para dominar profundamente a MP é preciso entender bem a matemática e as propriedades das formas, pois cada elemento geométrico é definido por um conjunto de regras algébricas e propriedades que não mudam. Como exemplo disso, temos a forma de um quadrado. Para parametrizá-lo digitalmente, é preciso deixar claro para o computador quais são suas propriedades invariantes, que façam com que esse polígono de quatro lados seja reconhecido como um quadrado: ter quatro lados iguais, todos os ângulos internos retos e lados opostos paralelos entre si. Essas condições, que não mudam, são o que diferenciariam o quadrado de outros tipos de quadriláteros (retângulos, losangos, etc), mesmo quando seus parâmetros variáveis (como a dimensão do lado, por exemplo) sejam alterados.

Essa relação entre parâmetros fixos e variáveis se estende a todos os elementos da modelagem paramétrica. Um parâmetro, nesse contexto, é justamente uma propriedade de valor variável que atua junto com definições invariantes, estabelecendo, assim, a base conceitual de um objeto. Romcy e Cardoso (2019) classificam os parâmetros nessas duas categorias: os fixos (restrições), que funcionam como regras imutáveis garantindo conformidade com requisitos técnicos e normativos; e os variáveis, que permitem explorar variações de soluções dentro do espaço delimitado por essas restrições. Esse jogo - entre as restrições e a flexibilidade criativa, é o que permite a integração de viabilidade construtiva e inovação, entre o

atendimento às definições imutáveis que lhes confere identidade e a possibilidade de variação formal (SACKS *et al.*, 2004).

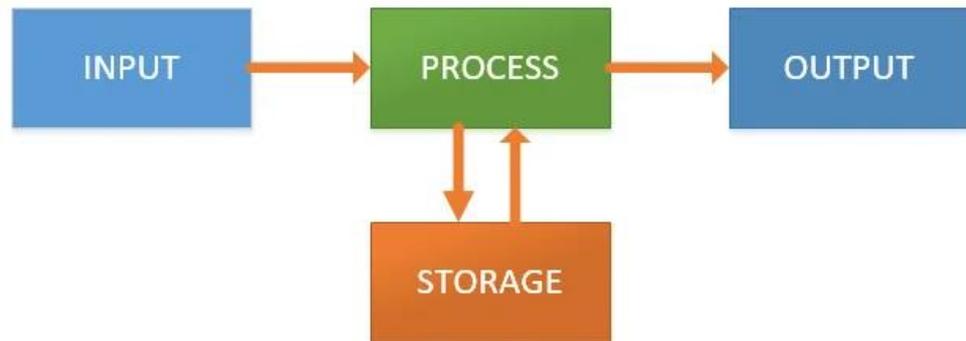
Dessa forma, Leach (2014) e Woodbury (2010) argumentam que a MP muda a prática projetual do "criar" para o "descobrir", transformando a tarefa de "criar modelos digitais" - de um ato de criar formas (*making of form*), para um processo de encontrar formas (*finding of form*). Essa abordagem reflete o pensamento paramétrico, onde os resultados vêm a partir da compreensão dos parâmetros e regras iniciais. Ou seja, o projetista não altera cada variação manualmente, mas define relações entre variáveis que controlam todo o sistema, que é alterado automaticamente. É por isso que dominar a modelagem paramétrica vai além da técnica - exige entender a lógica geométrica por trás dessas relações e como elas geram resultados coerentes.

Como comentado, a MP quando feita em ambientes BIM, vai além da geometria, integrando-se diretamente com os metadados do modelo. Quando combinada com a MA, essa junção cria sistemas altamente adaptáveis. A (MA) consiste na criação e na manipulação de modelos digitais através de algoritmos. Segundo Florio (2017), um algoritmo pode ser entendido como um conjunto de tarefas que, seguidas em um passo a passo lógico, orientam a resolução de algum problema. Quando o processamento das ações definidas no algoritmo é feito em um programa executável, tem-se a programação algorítmica, que pode ser textual ou visual (TEDESCHI, 2014). Como explica Tedeschi (2014), a MA permite manipular os *softwares* "por dentro", criando procedimentos a partir da própria linguagem computacional deles, ou seja, a partir dos algoritmos.

Antes de falar na definição de programação visual ou textual, é importante ter um entendimento sobre programação computacional (PC). A PC consiste em um processo de criar instruções (códigos, algoritmos ou *scripts*) em linguagens específicas para executar tarefas em programas executáveis no computador.

Em essência, todas as linguagens de programação compartilham a mesma lógica computacional: são sistemas que permitem pegar dados de entrada (*input*), processá-los conforme regras específicas (*process*), armazenar informações temporária ou permanentemente (*storage*) e, por fim, gerar saídas (*output*) que seriam o resultado de tudo (Silva, 2020). Os *outputs* podem ser desde textos (*strings*) até interfaces gráficas ou comunicação entre dispositivos. Esse fluxo - *input, process, storage, output* (Figura 2), mostra a versatilidade da programação computacional, onde cada etapa pode ser controlada e adaptada às necessidades do projeto.

Figura 2: Fluxo lógico da programação



Fonte: Walmir Silva, 2020

Blikstein (2008) destaca que o verdadeiro potencial da PC está em transformar o computador em uma extensão da nossa capacidade cognitiva, aumentando não apenas a produtividade, mas também a criatividade e a inovação. Como observa o autor, em um mercado cada vez mais competitivo, profissionais como engenheiros, arquitetos, economistas e cientistas podem - ao dominar essas ferramentas e a lógica por trás desse pensamento computacional, desenvolver modelos mais eficientes e resolver problemas complexos com mais facilidade.

Seguindo essa mesma lógica, a MA se divide em textual e visual. A textual acontece quando o usuário escreve as instruções em linhas de texto, com uma linguagem de código específica, como por exemplo o *Python*, *C#* ou *JavaScript*¹⁴. A modelagem feita com programação textual pede que o programador domine a sintaxe exata da linguagem escolhida, digitando por escrito todos os comandos que o computador interpreta e executa. Dessa maneira, o usuário consegue ter mais controle sobre as operações que está fazendo e, como consequência, tem mais liberdade para a criação de algoritmos mais complexos. No entanto, esse método pede uma curva de aprendizagem alta, já que cada linguagem possui suas próprias regras e lógicas que precisam ser memorizadas e aplicadas (MARCUS, 2019).

É importante ressaltar que o fluxo mostrado na Figura 02 não se limita a um único tipo de *software*, mas essa mesma linha de raciocínio está presente em diversas linguagens de programação, e cada uma com suas particularidades e finalidades. Na

¹⁴ O *JavaScript* é uma linguagem de programação textual que permite que você implemente itens mais complexos e dinâmicos em páginas da web. Enquanto o HTML e o CSS cuidam da estrutura e do estilo de uma página, o *JavaScript* pode ser utilizado quando for preciso exibir conteúdos atualizados em tempo real, como mapas interativos, gráficos animados ou formulários inteligentes.

Figura 3 é possível ver um exemplo de linguagem de programação textual feita em *JavaScript*, onde o autor cria um sistema que calcula automaticamente a média de idades de concurreseiros, demonstrando como as linguagens processam dados de entrada (*input*) e geram resultados (*output*).

Figura 3: Exemplo de programação textual em *JavaScript*

```

1 import java.util.Scanner;
2
3 public class Principal
4 {
5     public static void main(String[] args)
6     {
7         int numero;
8         int cont = 0;
9         int soma = 0;
10        int media;
11
12        System.out.println("Informe a idade dos concurreseiros:");
13        Scanner in = new Scanner(System.in);
14
15        while ((numero = in.nextInt()) >= 0)
16        {
17            soma += numero;
18            cont++;
19        }
20
21        in.close();
22
23        media = soma/cont;
24
25        System.out.println("\nA média de idade dos concurreseiros é de " + media + " anos." );
26
27    }
28
29 }

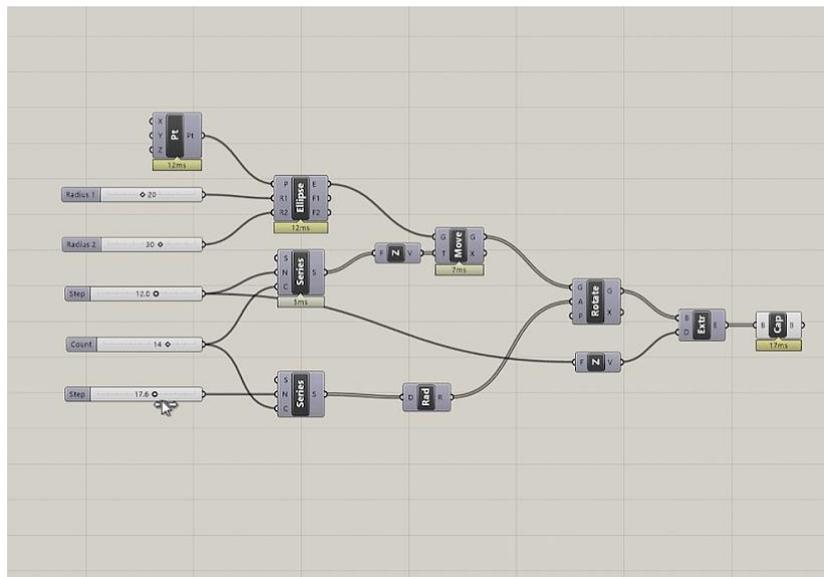
```

Fonte: Cristiane Neves, 2023

Já a PV é uma abordagem de desenvolvimento computacional que troca códigos textuais por interfaces gráficas interativas, onde os algoritmos são feitos através da manipulação direta de elementos visuais (nós, blocos ou fluxogramas) e suas conexões. Essa abordagem é usada para tornar mais acessível a PC, principalmente para profissionais que já se utilizam da linguagem gráfica em sua comunicação. Isso porque, enquanto as linguagens textuais exigem um domínio mais preciso dos textos de programação, as abordagens visuais levam esses mesmos conceitos de *input - process - output* para elementos gráficos manipuláveis, criando ambientes mais acessíveis para programar e chegar aos resultados esperados. Segundo Maleki *et al.* (2022), essa transição do textual para o visual não é um abandono dos princípios da computação, mas sim uma nova forma de como esses conceitos são representados e manipulados. O mesmo fluxo lógico de programação textual, pode também ser feito visualmente através de nós e conexões, onde as variáveis e funções se transformam em elementos arrastáveis que é possível

“costurar” sua programação através da ligação de um bloco ao outro (Figura 4). Nesses ambientes gráficos, os usuários conectam blocos ou nós que representam funções, dados ou ações, criando fluxos lógicos de maneira intuitiva e visual. Isso, conseqüentemente, oferece uma interface mais acessível para a criação dos códigos esperados.

Figura 4: Exemplo de programação visual



Fonte: Danil Nagy, 2020

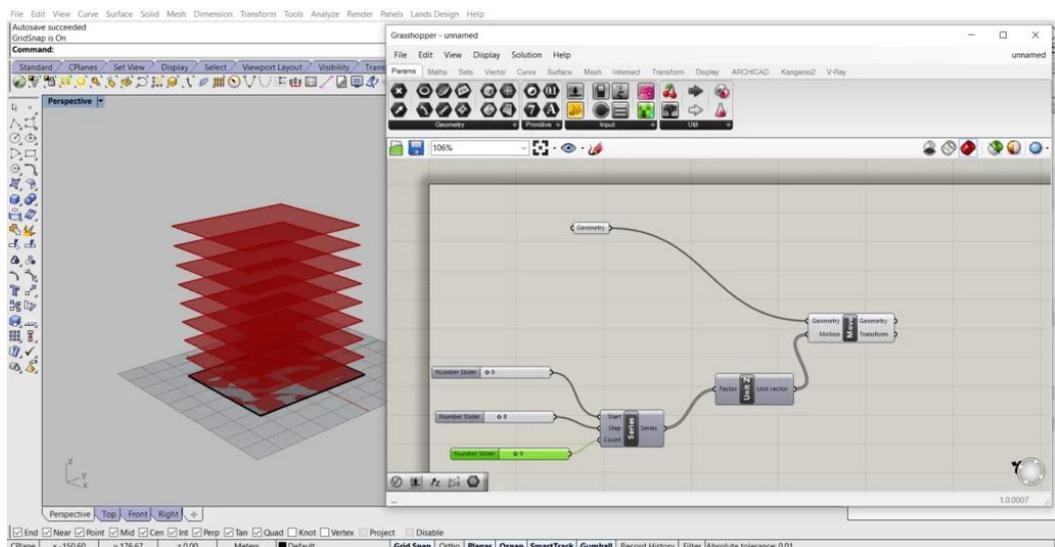
Brennan e Resnick (2012) argumentam que essa representação visual de algoritmos pode incentivar o pensamento espacial e sistêmico - que muitas vezes não é tão explorado na programação textual tradicional. Por exemplo, quando uma pessoa manipula visualmente um algoritmo e vê em tempo real as conseqüências que acontecem a partir disso, cria-se uma relação entre a intuição e a lógica por trás desse pensamento que fará chegar ao resultado - processo que Case *et al.* (2023) identificaram como característico do pensamento algorítmico projetual.

Segundo Myers *et al.* (2006), essa visualização é muito importante em domínios onde o pensamento espacial é central, como no caso da arquitetura, onde relações geométricas podem ser mais facilmente entendidas através de conexões visuais do que por descrições em forma de texto. Alguns ambientes de programação visual permitem a inserção de linguagens textuais dentro dos fluxos visuais, combinando esse jeito mais acessível das interfaces gráficas com a precisão do código textual e a liberdade maior que ele proporciona. Entre os muitos benefícios dessa abordagem

está a capacidade de criar modelos consistentes e adaptáveis, que respondem automaticamente a alterações nas informações de entrada. Isso não só pode diminuir o tempo gasto em ajustes manuais, mas também melhorar a precisão dos elementos modelados.

Entre alguns ambientes mais conhecidos para programar visualmente, podemos citar o *Dynamo*, da *Autodesk*, e o *Grasshopper*, da Robert McNeel & Associates. O *Grasshopper* é uma ferramenta de PV bastante usada e integrada ao *software Rhinoceros* (um *software* de modelagem paramétrica que se baseia em NURBS¹⁵). A interface do *Grasshopper* acontece em uma tela chamada "Canvas", onde os componentes de comando, representados por ícones visuais, são conectados através das entradas (*inputs*) e das saídas (*outputs*), formando um fluxo lógico do algorítmico, e facilitando a visualização e a manipulação dos dados. A integração com as funcionalidades do *Rhinoceros* e sua abordagem visual faz com que o *Grasshopper* seja uma escolha popular entre os arquitetos, designers e engenheiros para a elaboração de estudos, simulações e representações precisas em projetos arquitetônicos e de design.

Figura 5: Exemplo de programação visual feita no *software Grasshopper*

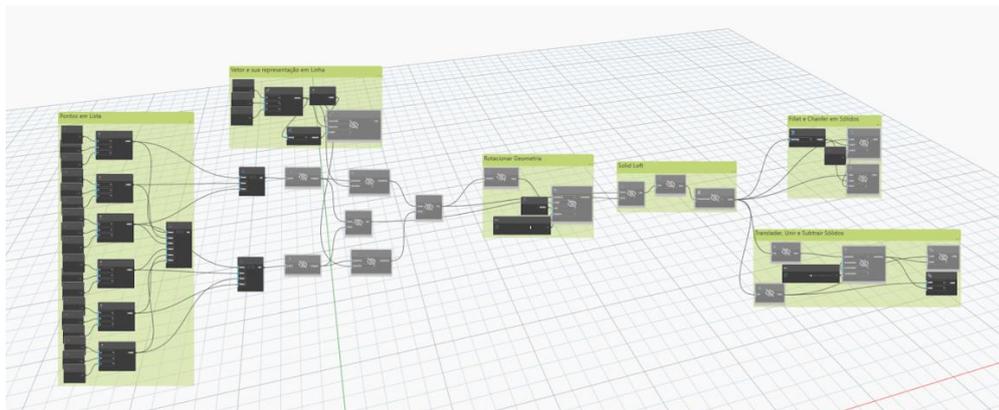


Fonte: *SPBIM*, arquitetura, engenharia e construção

¹⁵ NURBS (*Non-Uniform Rational B-Splines*): Método matemático para representação de superfícies e curvas em modelagem 3D, que permite formas orgânicas e precisas através do controle de pontos, pesos e graus de curvatura.

Assim como o *Grasshopper*, o *Dynamo* é um ambiente de PV de código aberto que possibilita aos projetistas a personalização dos fluxos de trabalho. A diferença é que, neste caso, o *software* já vem integrado a uma plataforma BIM - o *Revit*, da *Autodesk*. Ele fornece um ambiente intuitivo para a modelagem paramétrica, permitindo que os projetistas criem geometrias complexas de forma ágil e precisa, visualizando as modificações feitas em seus parâmetros e os impactos disso em segundos na forma geométrica. A MA no *Dynamo* é feita por meio da conexão de nós visuais que representam funções, parâmetros e dados, permitindo que alterações em variáveis de entrada sejam refletidas automaticamente no modelo final. Isso permite uma flexibilidade mais para explorar diferentes soluções. Essa abordagem permite a exploração de diversas configurações com base em um conjunto de regras, ampliando as possibilidades de design e promovendo uma prática mais investigativa e dinâmica.

Figura 6: Exemplo de programação visual feita no Dynamo



Fonte: Acervo pessoal da autora

Em resumo, a MA traz consigo essa capacidade de criar geometrias personalizadas que seriam difíceis ou quase impossíveis de se fazer com métodos convencionais. Tais geometrias são possibilitadas a partir da tradução de operações matemáticas avançadas em algoritmos desenvolvidos em uma interface gráfica intuitiva. O benefício claro está na precisão no controle dessas geometrias não convencionais: curvas complexas, padrões iterativos ou sistemas adaptativos podem ser criados e modificados em tempo real. Isso elimina o trabalho de ajuste manual ponto a ponto, enquanto mantém o rigor matemático necessário para fazer os projetos executivos (TEDESCHI, 2014; AUTODESK, 2023).

Por fim, como já mencionado, a PV pode ajudar significativamente na automatização da modelagem de projetos, e no capítulo a seguir serão identificadas e classificadas algumas atividades do processo de modelagem que podem ser auxiliadas por essa abordagem.

2 A MODELAGEM PARAMÉTRICA COMO FERRAMENTA DE AUTOMATIZAÇÃO DO PROCESSO DE MODELAGEM DO PROJETO

Como comentado, a MP, quando potencializada pela PV, oferece ganhos significativos no fluxo de trabalho da AECO. Seu uso na concepção projetual pode ampliar as possibilidades criativas, enquanto as automações no processo de modelagem de projeto podem otimizar tempo e recursos no cotidiano profissional. Este estudo concentra-se especificamente nesse segundo aspecto: explorar automações aplicáveis à modelagem de projetos, com base em experiências documentadas na literatura.

Essas automações podem ser implementadas através de rotinas parametrizadas em *softwares* de modelagem algorítmica. Tais rotinas podem ter finalidades diversas e, assim, ser classificadas de diferentes maneiras, dentro das atividades feitas na modelagem BIM. A criação de objetos BIM, como portas, paredes, pisos e equipamentos, por exemplo, pode ser significativamente automatizada através dessas rotinas, em vez de modelar, e até inserir no modelo, cada elemento manualmente.

Objetos BIM são componentes modelados digitalmente em ambiente BIM que integram sua forma geométrica com informações não geométricas (metadados) e que podem ser parametrizados. Sendo assim, as rotinas podem gerenciar tanto a informação geométrica, quanto os metadados desses objetos.

Outra atividade desafiadora é a modelagem de formas complexas. Seja em plataformas BIM ou em *softwares* convencionais, essas formas demandam muito trabalho manual ou seria impossível de se conseguir. Como visto no Capítulo 1, a PV se mostra especialmente eficaz em projetos com geometrias não convencionais, como fachadas curvilíneas ou estruturas complexas, onde cada elemento constituinte precisa manter relações geométricas precisas com os demais - seja para garantir encaixes entre componentes, respeitar as tolerâncias construtivas ou atender a requisitos específicos.

A documentação em um projeto, presente em todas as fases desde o estudo preliminar até o projeto executivo, também pode se beneficiar, e muito, das automações feitas com a PV. Como comentado na Introdução, a documentação é um uso BIM que engloba diversas atividades, indo da criação de vistas bidimensionais

dos modelos (plantas, cortes e fachadas) até a inserção de simbologias exigidas pelas normas de documentação de projetos - como nomeação de ambientes, cotagem, colocação de eixos, identificação de esquadrias e inserção de especificações técnicas. Esses elementos, como os eixos por exemplo (linhas de referência para orientar a localização de elementos no projeto) podem ter sua numeração e organização totalmente automatizadas através de rotinas parametrizadas. Além disso, todo o fluxo documental citado pode ser otimizado e padronizado. Na prática observa-se que algumas rotinas desenvolvidas em PV são capazes de criar automaticamente as vistas de um modelo e atualizar dinamicamente legendas conforme o modelo evolui, garantindo a sincronização entre todas as representações gráficas do projeto.

Outra atividade corriqueira em um fluxo de trabalho BIM é a extração e gerenciamento de metadados do modelo geométrico. A automatização dessa atividade pode abrir novas possibilidades para análises e tomada de decisão. Com uma rotina capaz de extrair automaticamente os metadados para planilhas financeiras, é possível automatizar, também, cálculos de quantitativos, custos e prazos com base nos elementos modelados, por exemplo. Isso permite não apenas acompanhar o orçamento em tempo real, mas também simular diferentes cenários ao variar parâmetros como tipos de materiais ou métodos construtivos.

A extração de metadados através de PV também pode permitir a verificação automática da conformidade desses com normas e regulamentos. Ao cruzar as informações do modelo com as normas de projetos, essas rotinas podem analisar parâmetros como acessibilidade, desempenho térmico ou segurança estrutural, identificando não conformidades antes mesmo da etapa de execução, sendo, assim, uma grande ferramenta de validação técnica.

Diante dessas atividades de modelagem BIM identificadas - desde a criação de objetos e formas complexas até a documentação e gestão de metadados - observa-se que a PV pode automatizar várias etapas do fluxo de modelagem. Essas possibilidades de automação podem ser classificadas em algumas categorias conforme sua natureza e objetivos.

A primeira classificação é a criação de objetos BIM. Como comentado, os Objetos BIM para projetos arquitetônicos e urbanos são diversos. Neste trabalho, a pesquisa se concentra especificamente na criação de elementos construtivos como paredes, pisos e pilares. Esse tipo de estudo foi escolhido por ser um problema comum em escritórios e empresas de arquitetura e engenharia: a necessidade de

gerar e atualizar elementos estruturais de forma ágil, mantendo consistência com critérios pré-definidos (como dimensões, posicionamento e propriedades materiais que sempre vão ter em um modelo).

Na segunda classificação tem-se a documentação automática de projetos. Esta, abrange desde a geração inteligente de plantas, cortes e fachadas, até a gestão dinâmica de toda a simbologia técnica exigida - como cotas, eixos, legendas e identificações.

Por fim, a terceira classificação é a gestão de metadados. A automatização de tarefas dessa natureza permite não apenas a rápida e precisa extração de informações do modelo para análises quantitativas e orçamentárias, mas também a verificação da conformidade do projeto com normas técnicas, como as de acessibilidade, ventilação, desempenho e segurança, entre outras.

Estas categorias, embora não sejam as únicas, foram identificadas ao longo do trabalho e terão algumas de suas aplicabilidades exploradas mais a fundo. Elas representam tanto os processos mais repetitivos na prática profissional, quanto aqueles onde a PV pode trazer ganhos mais expressivos - aumentando a produtividade, melhorando a gestão de informações e elevando a eficiência nos projetos.

Para exemplificar algumas dessas tarefas do ato da modelagem BIM, são apresentados dois estudos de caso, a partir de outros trabalhos, mostrando como a PV pode auxiliar em automatizações no processo de modelagem BIM.

2.1 Criação de Objetos BIM: Detalhamento de Paredes de Reservatórios de Concreto Armado usando a Ferramenta *Dynamo*

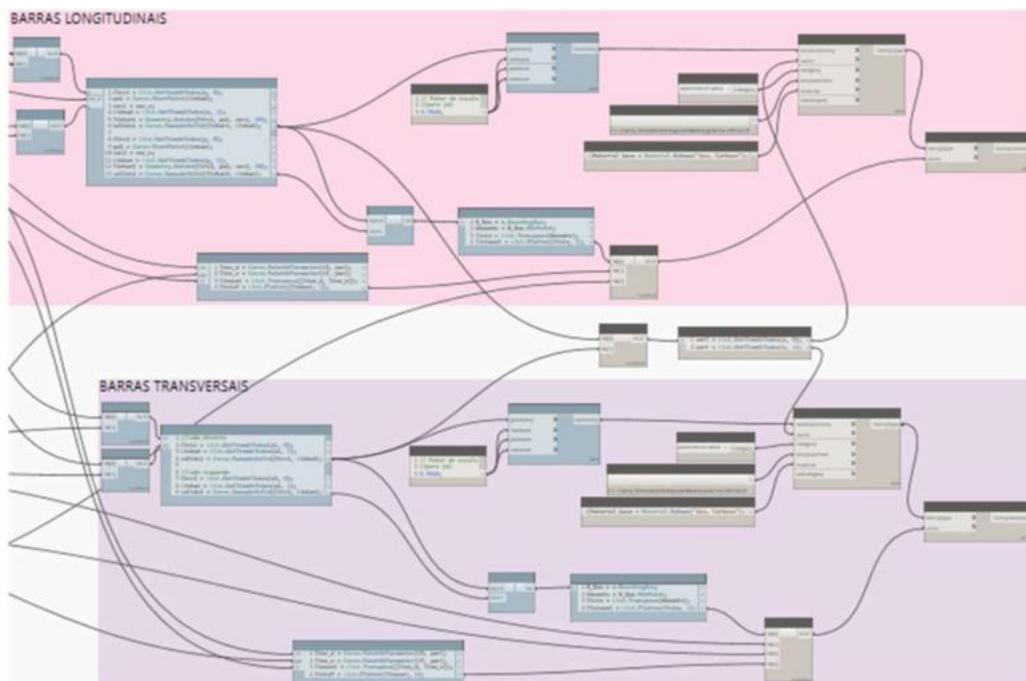
O trabalho de Moraes (2023) desenvolve uma rotina em *Dynamo* para automatizar o detalhamento de armaduras em paredes de reservatórios de concreto armado, garantindo o cumprimento de requisitos normativos da NBR 6118:2014. A autora mostra que a rotina assegura aspectos de cobrimento mínimo de 3 cm (distância entre a face do concreto e as armaduras, exigido para estruturas enterradas em contato com o solo); a distribuição uniforme das barras, respeitando espaçamentos máximos; e as armaduras mínimas para resistir aos esforços de flexão (quando uma força aplicada a um objeto faz com que ele curve ou dobre). Como o

Revit não possui ferramentas nativas para esse dimensionamento estrutural conforme as normas brasileiras, a solução proposta pela autora é eliminar a necessidade de inserção manual dessas armaduras, reduzindo erros comuns como espaçamentos irregulares ou desrespeito ao cobrimento, enquanto mantém a conformidade com os critérios técnicos.

Entendendo isso, a rotina começa filtrando as paredes modeladas no *Revit* com base em propriedades como: o tipo de família - "Parede de concreto armado de 15 cm"; a espessura (definida em 15 cm para atender ao mínimo da NBR 6118); e o material (concreto armado). Em seguida, extrai dados como altura, comprimento e deslocamento da base para calcular a distribuição das armaduras.

Depois, para aplicar a as regras da NBR 6118, a rotina se divide em três etapas. Primeiro, ela extrai os metadados do projeto (como a classe de agressividade ambiental) diretamente dos parâmetros das paredes modeladas. Em seguida, calcula e insere automaticamente o cobrimento de 3 cm, posicionando as armaduras a essa distância das faces do concreto, e a armadura mínima de flexão, distribuindo as barras uniformemente conforme os esforços. Esse processo é feito através de rotinas que convertem os requisitos normativos citados, nas regras paramétricas, e qualquer alteração nas dimensões da parede atualiza automaticamente o detalhamento das armaduras.

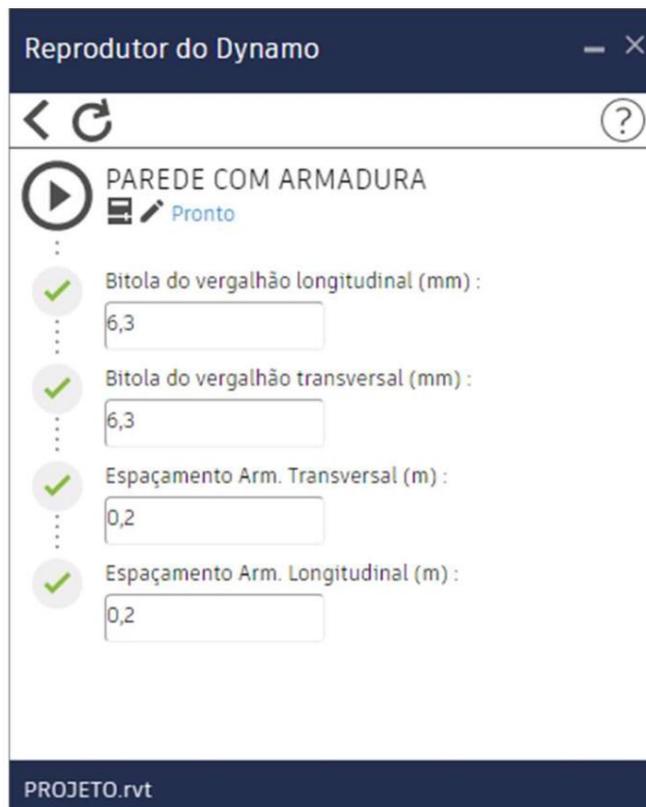
Figura 7: Rotina do estudo



Fonte: Moraes, 2023. Disponível no trabalho original.

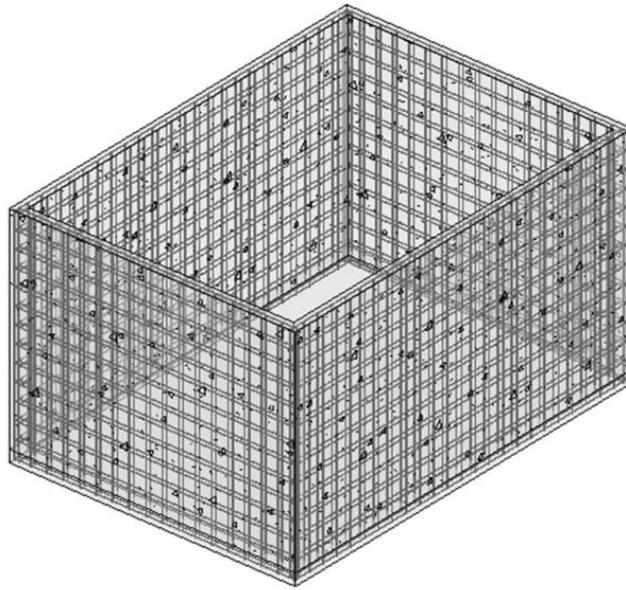
A autora inseriu a rotina final na Janela do Reprodutor do *Dynamo* - o *Dynamo Player*. Dessa forma, o usuário só precisa inserir no *Dynamo Player* a bitola (ex.: 6,3 mm) e o espaçamento (ex.: 20 cm), e o sistema gera automaticamente as barras longitudinais e transversais, posicionando elas corretamente ao longo das paredes (Figura 8). Se as dimensões da parede forem alteradas, as armaduras se ajustam automaticamente.

Figura 8: Reprodutor do *dynamo* para a rotina



Fonte: Moraes, 2023. Disponível no trabalho original.

Figura 9: Paredes do reservatório com armaduras introduzidas



Fonte: Moraes, 2023. Disponível no trabalho original.

Essa rotina também incorpora verificações automáticas das exigências normativas. Dentre elas estão o cômputo de áreas de aço mínimas e máximas e espaçamentos permitidos. Esse cômputo é feito a partir da extração dos metadados das geometrias que continham essas informações, garantindo que as soluções geradas sempre estejam em conformidade com as normas técnicas. Segundo os resultados apresentados pela autora, essa abordagem permitiu uma redução de aproximadamente 80% no tempo de detalhamento manual, além de garantir a conformidade com os requisitos da NBR 6118. O estudo demonstrou como a integração entre MP e PV pode trazer ganhos significativos de produtividade e precisão para o fluxo de trabalho em projetos estruturais.

2.2 Documentação e pré-dimensionamento de projeto: Automatização no desenvolvimento de projetos sustentáveis de engenharia via metodologia BIM

Silva *et al.* (2024) propuseram um conjunto de rotinas em *Dynamo* para automatizar algumas etapas de modelagem estrutural de edificações em concreto armado. Essas vão desde a documentação, como a criação de eixos, numeração de

elementos e pré-dimensionamento, até o detalhamento técnico final, conforme a NBR 6118:2023.

Figura 10: Lógica para produção dos scripts



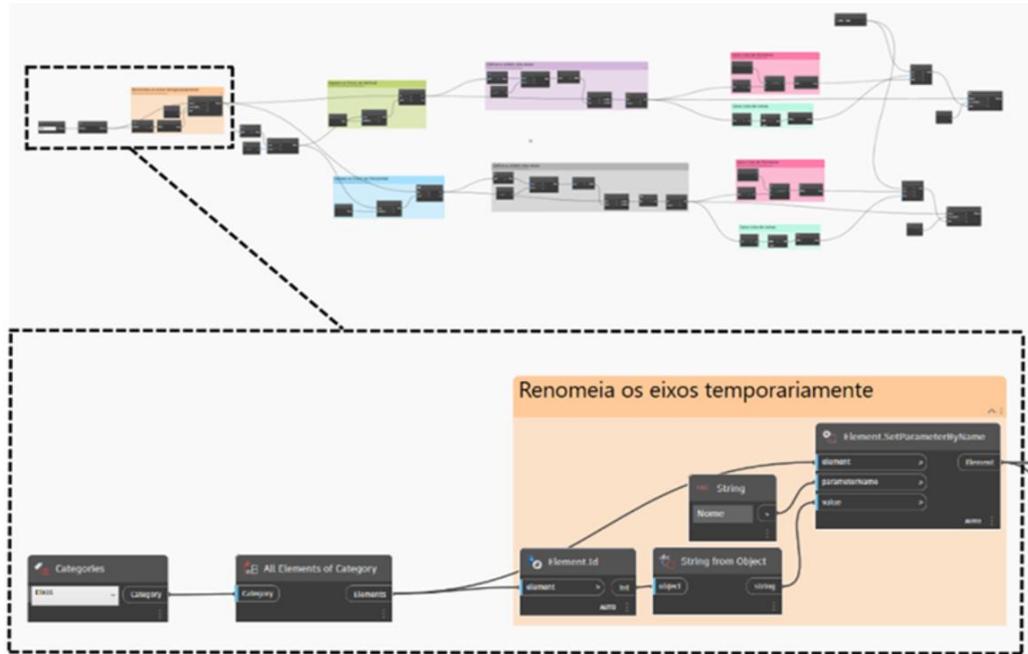
Fonte: Silva *et al* (2024). Disponível no trabalho original.

A rotina começa com os eixos estruturais sendo identificados: os verticais recebem números (1, 2, 3) e os horizontais, letras (A, B, C), organizados de acordo com suas posições. A figura 11 mostra a rotina completa, no grupo laranja, está a demonstração da parte da rotina que renomeia os eixos. Os Pilares e as fundações são numerados sequencialmente (por exemplo, P1, P2) com base em suas coordenadas, enquanto as vigas contínuas - vigas únicas que passam por mais de um vão, são agrupadas sob uma mesma numeração, com letras diferenciando cada segmento (como V101a, V101b).

Para o pré-dimensionamento, a rotina relaciona os elementos do modelo com critérios técnicos, como a relação entre o vão livre (L) e a altura da viga ($L/10$). Assim, uma viga com 5 metros de vão recebe automaticamente uma altura preliminar de 50 cm. Esses cálculos consideram dados de entrada como bitolas, cargas e geometria dos elementos, gerando planilhas em formato .xls com valores padronizados.

Por fim, a rotina cria vistas automaticamente: plantas e cortes para fundações, vistas frontais e cortes por pavimento para pilares, e cortes longitudinais e transversais para as vigas. Os resultados apresentados pelos autores revelam ganhos em produtividade no processo de modelagem de projeto (com redução de até 70% no tempo).

Figura 11: Rotina do estudo



Fonte: Silva *et al* (2024). Disponibilizado no trabalho original.

Figura 12: Demonstração dos eixos criados



Fonte: Silva *et al* (2024). Disponibilizado no trabalho original.

Os estudos de caso apresentados neste capítulo mostram como a PV pode automatizar algumas tarefas repetitivas da modelagem e fazer com que sejam

processos mais rápidos e eficientes. No capítulo a seguir, serão exploradas rotinas práticas baseadas na classificação estabelecida anteriormente: criação de objetos BIM, documentação automática de projetos e gestão de metadados (com extração de dados para análise e verificação de conformidade normativa). A exploração dessas rotinas tem como objetivo demonstrar na prática como a lógica paramétrica e algorítmica podem automatizar tarefas.

3 EXPLORAÇÃO E DESENVOLVIMENTO DE AUTOMATIZAÇÕES RELACIONADAS À PROCESSOS DE MODELAGEM DE PROJETO, SOB A LÓGICA PARAMÉTRICA E ALGORÍTMICA

Este capítulo detalha as explorações sobre as automações realizadas na pesquisa, organizadas em três partes. A primeira foca na criação automatizada de objetos BIM. O foco está na aplicação prática para geração de elementos construtivos como pilares, paredes e pisos, demonstrando como conectar bases de dados externas, como planilhas em *excel*, ao ambiente de modelagem e quais os benefícios é possível ter a partir desse fluxo de trabalho. A segunda, trata da automação para documentação de projeto, criando os níveis do projeto e suas respectivas pranchas. Por fim, a terceira parte tem foco na gestão de metadados, fazendo a verificação automática das áreas de ventilação natural, seguindo os critérios estabelecidos pela NBR 15.575-4:2021.

3.1 Criação de objetos BIM: pilares, paredes e pisos

A rotina inicia com a criação parametrizada de pilares e se expande para gerar paredes e pisos de forma integrada, estabelecendo relações entre os elementos modelados. Ela está vinculada a uma planilha externa contendo as coordenadas, dimensões, identificadores, área e volume que serão considerados. Qualquer ajuste feito nos dados originais desta planilha modifica automaticamente o modelo. Essa rotina se mostra muito valiosa em projetos que pedem muitas revisões e alterações, sem a necessidade de refazer trabalhos manuais de atualização da forma geométrica desses elementos, garantindo que as alterações sejam consistentes no modelo.

O passo a passo da automação da criação desses objetos BIM está descrito a seguir.

3.1.1 Pilares

A primeira parte da rotina consiste na criação automatizada de pilares a partir de dados de planilha *Excel*. Nesta planilha, cada coluna corresponde a um parâmetro específico dos pilares. As informações contidas são: as coordenadas X e Y, que

definem o posicionamento de cada pilar no projeto; as dimensões deles: largura, comprimento e altura; identificadores únicos para cada elemento (p1, p2...); além de dados como área das superfícies e volume. Esta planilha servirá como base para todo o processo subsequente.

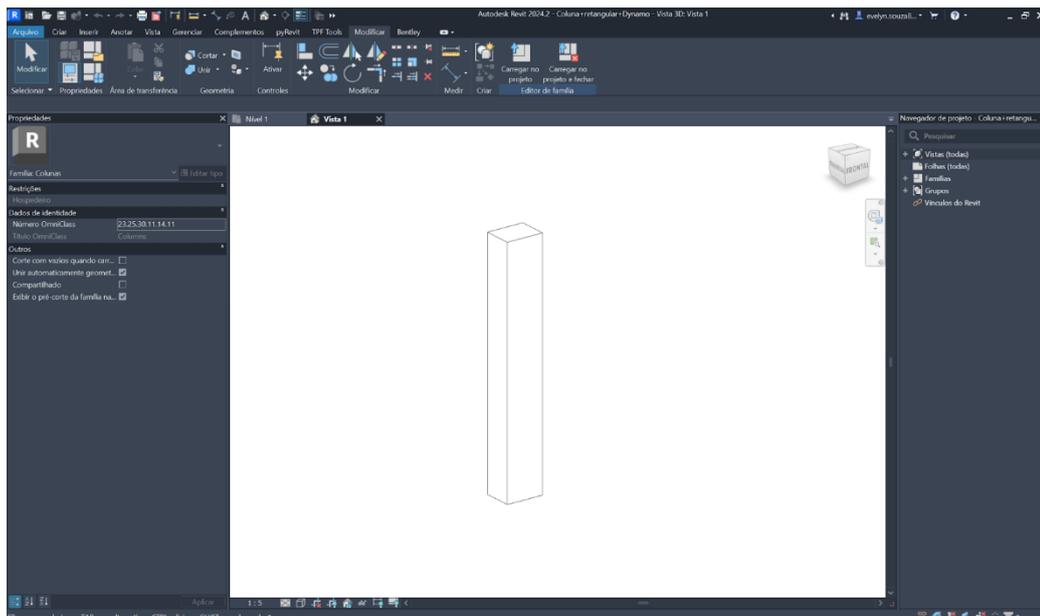
Figura 13: Planilha adotada no estudo 01

Pilar	Largura	Comprimento	X	Y	Área de Superfície	Volume
P1	0,3	0,3	0	2	4,8	0,45
P2	0,5	0,3	3	2	6	0,75
P3	0,5	0,3	6	2	5,4	0,6
P4	0,4	0,3	9	2	5,4	0,6
P5	0,5	0,3	12	2	7,8	1,2
P6	0,8	0,3	0	6	9,6	1,92
P7	0,5	0,3	3	6	7,2	1,05
P8	0,4	0,3	6	6	4,8	0,48
P9	0,3	0,3	9	6	4,8	0,45
P10	0,5	0,3	12	6	6	0,75

Fonte: A autora, adaptado de NeuroBIM Arquitetura e Engenharia (2025)

Antes de começar a desenvolver a rotina de fato, é preciso que um arquivo de família de pilar seja carregado no *Revit*. Para este estudo, foi utilizada a família de pilar mostrada na Figura 14:

Figura 14:Família de pilar utilizada

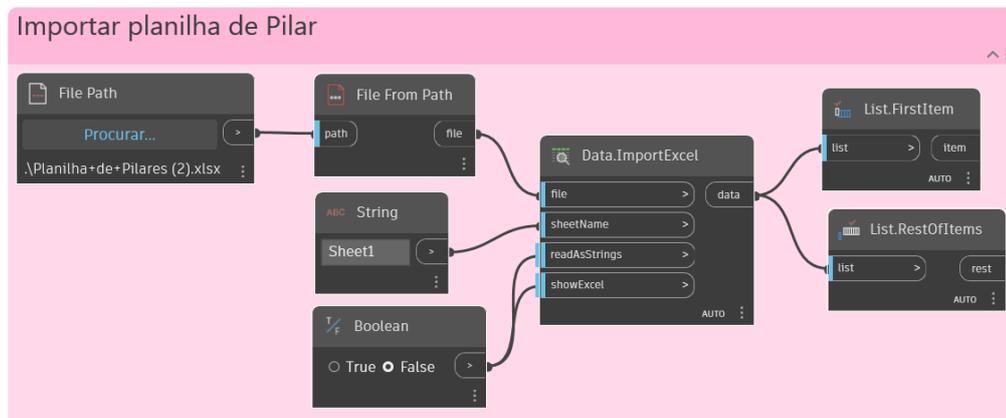


Fonte: A autora

Após carregar a família do pilar, abre-se a interface do *Dynamo* para importação dos dados da planilha inicial. O processo de importação em si já é pensado para ser flexível, pois o caminho do arquivo da planilha pode ser atualizado se for necessário, apenas selecionando a nova pasta onde o arquivo estaria eventualmente, para que os dados atualizados sejam transferidos para a interface do *Dynamo* novamente.

Durante o processo de importação, a rotina identifica qual aba da planilha contém as informações, convertendo esses valores em listas, como mostra a Figura 15.

Figura 15: Importação da planilha de pilar



Fonte: A autora

Com os dados da planilha disponíveis, a rotina passa para a criação dos pilares no modelo. Primeiro, as coordenadas X e Y são transformadas em pontos no modelo – esses pontos marcam onde cada pilar será colocado.

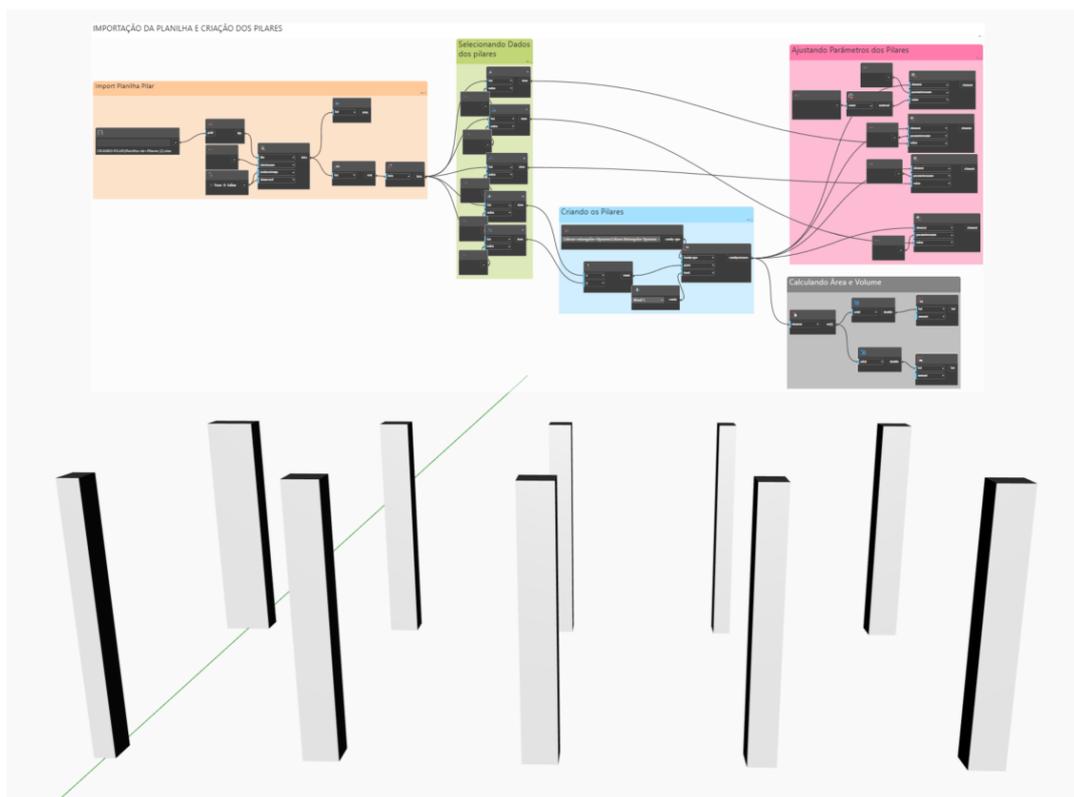
Em seguida, utiliza-se a geometria da família de pilar que foi carregada no *Revit* anteriormente, posicionando uma instância dela em cada ponto gerado pelas coordenadas da planilha. Por fim, a rotina ajusta a largura e comprimento de cada pilar, aplicando os parâmetros da planilha: as dimensões de cada um, a identificação (P1, P2, P3...) e o material.

Na Figura 16 é possível ver a rotina após sua execução dividida em subpartes (grupos com cores distintas). O grupo laranja é responsável por importar a planilha de pilares iniciais; o grupo verde seleciona os dados dos pilares - como o nome, largura, comprimento etc., para fazer a criação dos pilares propriamente ditos; essa criação dos pilares é feita a partir do grupo em azul, que, tendo como base a família de coluna

inicial de pilar, os cria no nível 01 do modelo; em seguida, o grupo em rosa ajusta os parâmetros dos pilares, aplicando o material "concreto moldado *in loco*" a eles e; por fim, o grupo cinza é responsável por criar a visualização das geometrias dos pilares no *Dynamo* a partir do nó "*Element.Geometry*".

Dessa forma, o processo de modelar manualmente pilar por pilar é substituído pela criação de todas as colunas instanciadas de uma só vez, posicionando-as com precisão de acordo com as coordenadas fornecidas na planilha. Além disso, qualquer alteração nos dados de entrada é automaticamente atualizada para todas as colunas. Esse é um recurso particularmente valioso em projetos de médio e grande porte, onde a criação manual de muitos elementos similares toma muito tempo e é mais propenso a erros.

Figura 16: Criação dos pilares a partir da planilha



Fonte: A autora

3.1.2 Paredes

Nessa rotina, o objetivo é modelar paredes conectadas aos pilares previamente criados. A princípio, é identificado o ponto central de cada pilar, que serve como base para traçar uma linha que representa o perímetro das paredes.

Os pontos centrais dos pilares são então conectados, formando a linha do perímetro. As extremidades desta linha definem os limites que as paredes serão modeladas¹⁶.

A criação das paredes é feita a partir desse traçado como dado de entrada principal (com o node "*Wall.ByCurveAndLevel*"), além dos seguintes *inputs*:

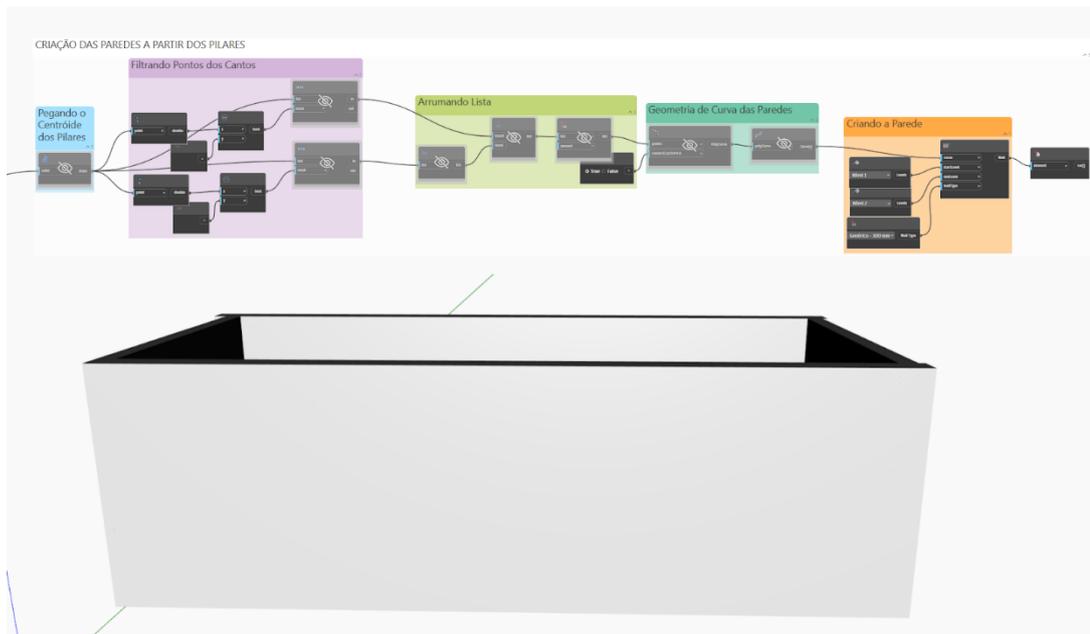
- Níveis: inferior (1) e superior (2)
- O tipo de parede desejado

Na figura 17 é possível ver as paredes criadas e integradas aos pilares iniciais após a execução da rotina. A integração ocorre através da conexão entre o centro de cada pilar e as linhas que representam o perímetro das paredes, que são traçadas passando por esses pontos. Dessa forma, qualquer modificação nos pilares reflete automaticamente nas paredes. Na rotina, a tarefa executada no grupo em azul é a identificação dos centroides dos pilares, enquanto o grupo em roxo filtra os pontos dos cantos dos pilares. Nos grupos verdes, o claro é responsável por organizar esses dados em listas, enquanto o escuro define as curvas onde as paredes serão passadas para alimentar o grupo laranja, que efetivamente cria as paredes por meio do nó "*Wall.ByCurveAndHeight*".

Com a rotina funcionando, e integrada à etapa de criação dos pilares, qualquer ajuste no layout pode ser rapidamente realizado. Isso é feito como a alteração dos pontos de referência e executando novamente a rotina no *Dynamo*, onde é gerado um novo arranjo de paredes sincronizado com os pilares previamente modelados.

¹⁶ Em programação, o ponto inicial e final de qualquer curva são acessados pelos parâmetros 0 e 1. Onde 0 representa o início da curva e 1 representa o final. Essa convenção foi utilizada para identificar as extremidades do perímetro.

Figura 17: Criação das paredes a partir dos pilares



Fonte: A autora

3.1.3 Pisos

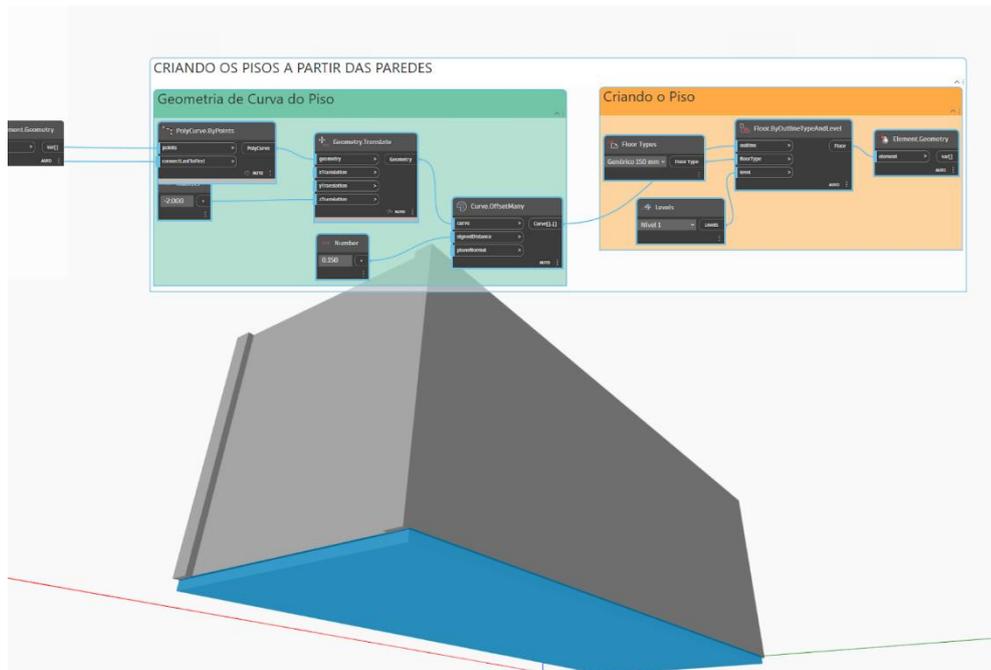
A criação dos pisos no *Revit*, a partir do *Dynamo*, é realizada utilizando três *inputs* principais: a linha que define o seu perímetro, o tipo de piso que vai ser usado e o nível de inserção.

A rotina inicia com a obtenção da linha que representa o perímetro, gerada a partir da delimitação das paredes já criadas na rotina anterior. Para garantir o alinhamento com as faces externas das paredes, aplica-se um parâmetro de deslocamento calculado como metade da espessura delas. Esse valor é utilizado para poder expandir o perímetro do piso, fazendo com que seus limites coincidam com as faces externas das paredes, sendo assim, quando o tipo de parede for alterado (e consequentemente sua espessura modificada), o parâmetro de deslocamento se ajusta automaticamente, mantendo sempre a relação geométrica entre as paredes e os pisos.

Na Figura 18 é possível ver a geometria dos pisos criada (destacada em azul). O grupo verde da rotina é responsável por processar a curva do perímetro, enquanto o grupo laranja de fato faz a criação do piso através do nó "*Floor.ByOutlineTypeAndLevel*", que recebe como inputs:

- O contorno obtido das paredes
- O tipo de piso
- O nível de inserção (Nesse caso, nível 1, mantendo a relação com as paredes e pilares)

Figura 18: Criação do piso



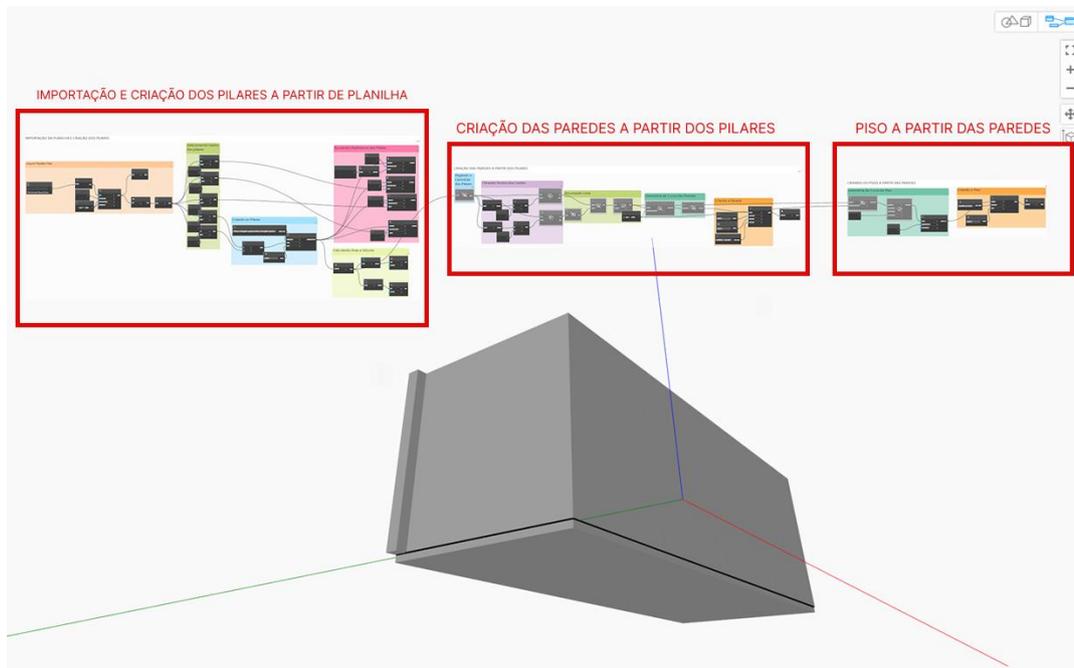
Fonte: A autora

É importante ressaltar que embora o processo automatizado cubra a maioria dos cenários, existem situações específicas. Exemplos disso são os encontros incomuns com pilares isolados ou geometrias complexas, que podem precisar de ajustes manuais pontuais. Essa limitação, contudo, é compensada pelos benefícios do fluxo todo - desde a redução de tempo de modelagem, a eliminação de erros de compatibilidade dimensional, e a atualização instantânea quando o projeto sofre alterações.

Após o desenvolvimento das três rotinas, elas foram integradas em um único arquivo do *Dynamo*. A conexão entre os *inputs* dos elementos (pilares, paredes e piso) é essencial para deixar o modelo integrado e adaptável para diferentes cenários. Dessa forma, o *input* dos centroides gerados na modelagem dos pilares serve como base para definir o perímetro das paredes, que por sua vez é utilizado - com o devido deslocamento comentado anteriormente - para criar a geometria dos pisos.

Abaixo está a rotina final contendo a geometria dos pilares, em seguida as paredes, e por último o piso.

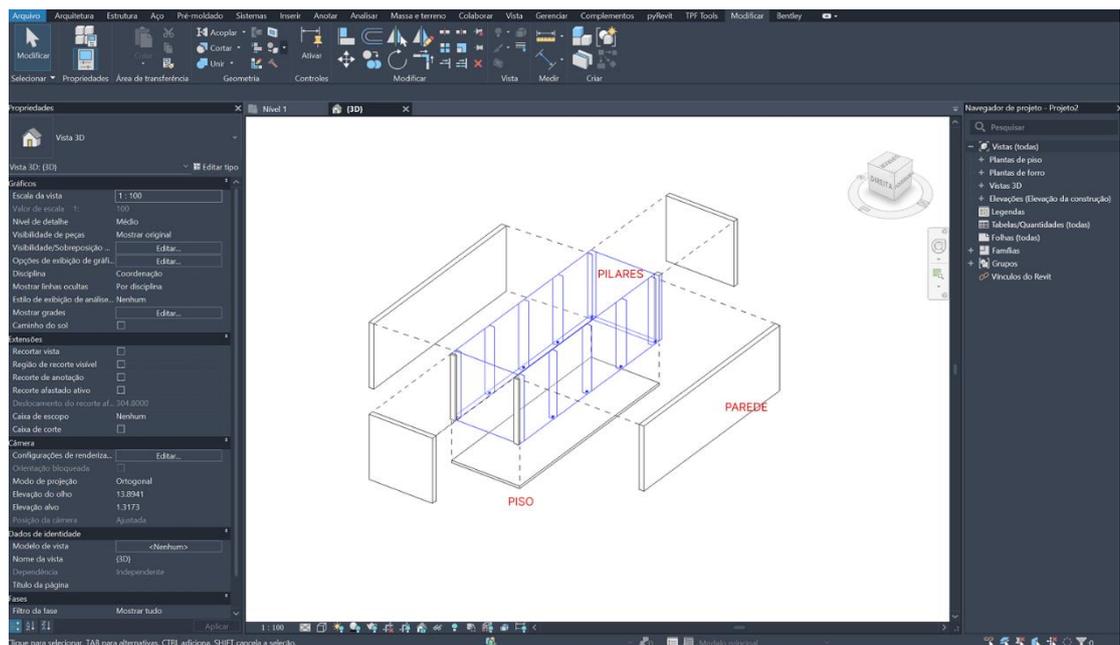
Figura 19: Rotina completa



Fonte: A autora

A Figura 20 ilustra essa integração através de uma vista explodida, mostrando todos os componentes criados e suas relações.

Figura 20: Modelo explodido no Revit



Fonte: A autora

O diferencial desse processo está na capacidade de adaptação às mudanças frequentes nos projetos. Desde a importação das informações das colunas na planilha, até a modelagem final do piso, cada etapa mostra como a automação é capaz de manter a coerência entre todos os elementos.

3.2 DOCUMENTAÇÃO DE PROJETO

Como explicado no Capítulo 02, a documentação de projeto, presente em todas as fases do desenvolvimento arquitetônico, exige a organização sistemática de plantas, cortes, elevações e detalhes, além da disposição de todas as anotações e em pranchas, conforme normas técnicas e padrões gráficos (NBR 6492:2021 Documentação técnica para projetos arquitetônicos e urbanísticos - Requisitos). Essa etapa é fundamental e, embora as plataformas BIM tenham avançado no processo de desenvolvimento de um modelo geométrico e informacional mais preciso, ainda demanda a repetição de tarefas e verificação constante, tornando-se um processo demorado e passível de inconsistências quando realizado manualmente.

Diante disso, são exploradas duas rotinas para automatizar e ajudar esse processo de documentação.

3.2.1 Criação de níveis automáticos

O desenvolvimento da primeira rotina para documentação técnica objetiva a criação automática de níveis de projeto, a partir de dados externos. Novamente as informações contidas em uma planilha *Excel* são usadas como dados de entrada. Nela, estão os nomes dos níveis e suas respectivas elevações, em metros. É importante perceber que a dimensão da elevação do nível é relativa ao nível 0, que seria o tamanho que cada pavimento vai adotar como pé esquerdo. Essa abordagem se mostra particularmente útil, por exemplo, em Planos de Execução BIM¹⁷ (PEB), onde os padrões do projeto são definidos antes mesmo do início da sua modelagem.

¹⁷ PEB (Plano de Execução BIM): Documento que estabelece diretrizes, protocolos e padrões para implementação de metodologia BIM em um projeto ou empreendimento.

Figura 21: Planilha adotada na rotina

Nível	Elevação
Garagem 03	-5,75
Garagem 02	-2,79
Garagem 01	0,17
Térreo	3,13
Pavimento 01	6,09
Pavimento 02	9,05
Pavimento 03	12,01
Pavimento 04	14,97
Pavimento 05	17,93
Pavimento 06	20,89
Pavimento 07	23,85
Pavimento 08	26,81
Pavimento 09	29,77
Pavimento 10	32,73
Cobertura	35,69

Fonte: A autora, modificado de NeuroBIM Arquitetura e Engenharia (2025)

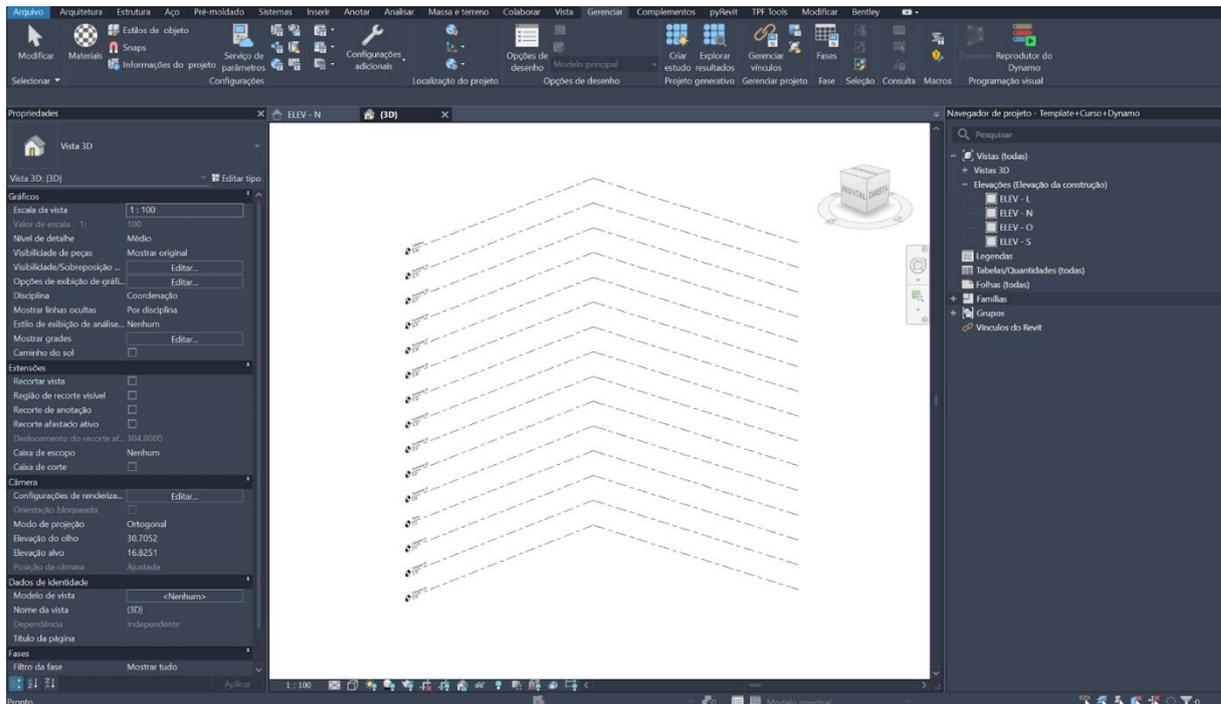
A rotina inicia com a importação desses dados da planilha através do nó central de importação (grupo verde, figura 22). Depois, esses dados são organizados em uma lista que separa cada nível e sua respectiva elevação (grupo rosa), e por fim, o nó “*Level.ByElevationAndName*” é utilizado para criar os níveis, cruzando as informações da lista criada (grupo azul).

Figura 22: Rotina do estudo



Fonte: A autora

Figura 23: visualização dos níveis gerados na plataforma



Fonte: A autora

É possível perceber que essa rotina permite a atualização dinâmica do modelo sempre que a planilha for modificada, assegurando a padronização e otimizando o tempo de configuração inicial, especialmente em projetos complexos com muito mais pavimentos.

3.2.2 Geração de pranchas automáticas

Ainda sobre a automatização da documentação de projetos, a segunda rotina foca na criação de pranchas para impressão. Esse processo envolve não apenas a geração das pranchas propriamente ditas, mas também a inserção automática dos nomes de cada vista, a partir de uma planilha e da inclusão das *viewports*¹⁸ associadas a cada prancha.

Dessa forma, o processo é iniciado com a leitura dos dados da planilha, onde informações como numeração, nomenclatura, vistas associadas e formatos de prancha (A0, A1 etc.) são organizadas em listas.

¹⁸ *Viewport* é um elemento que representa uma vista do modelo em uma folha de desenho.

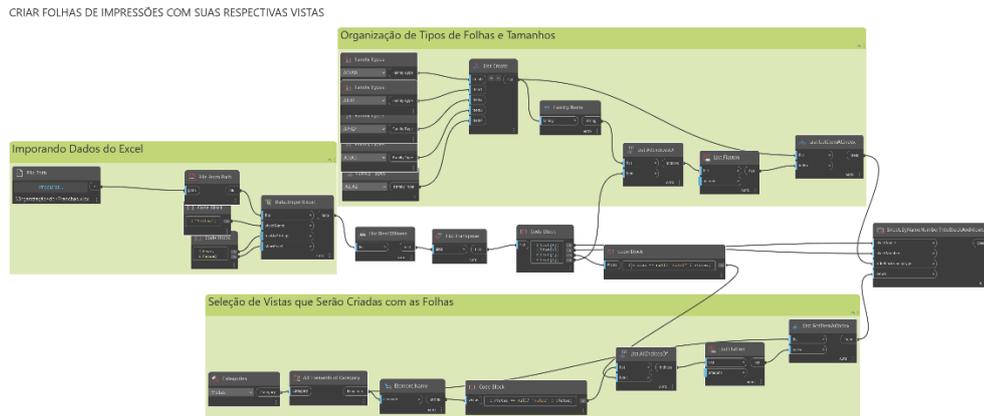
Figura 24: Planilha adotada na rotina

Número	Nome	Vistas	Tamanho
101	PLANTA DE PISO - G3	PLP - G3	A1
102	PLANTA DE PISO - G2	PLP - G2	A1
103	PLANTA DE PISO - G1	PLP - G1	A1
104	PLANTA DE PISO - TÉRREO	PLP - TER	A1
105	PLANTA DE PISO - PAV. TIPO	PLP - TIP	A1
106	PLANTA DE PISO - COBERTURA	PLP - COB	A1
201	PLANTA DE FORRO - G3	PLF - G3	A1
202	PLANTA DE FORRO - G2	PLF - G2	A1
203	PLANTA DE FORRO - G1	PLF - G1	A1
204	PLANTA DE FORRO - TÉRREO	PLF - TER	A1
205	PLANTA DE FORRO - PAV. TIPO	PLF - TIP	A1
301	ELEVAÇÃO NORTE	ELEV - N	A1
302	ELEVAÇÃO SUL	ELEV - S	A1
303	ELEVAÇÃO OESTE	ELEV - O	A1
304	ELEVAÇÃO LESTE	ELEV - L	A1
401	CORTES VERTICAIS	CORTE A	A1
402	CORTES HORIZONTAIS	CORTE B	A1
501	RENDERIZAÇÕES	RENDERIZAÇÕES	A2
601	QUANTITATIVOS	QUANT.	A3

Fonte: A autora, modificado de NeuroBIM Arquitetura e Engenharia (2025)

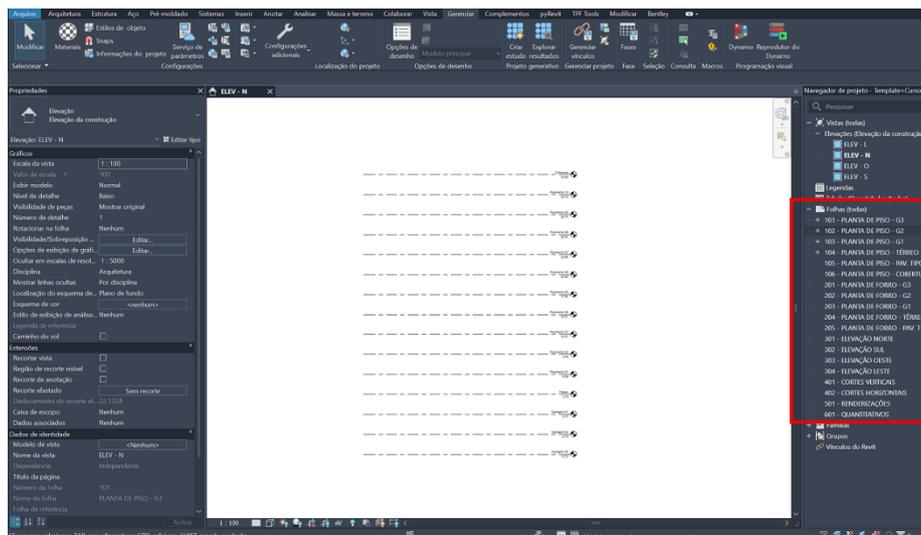
São quatro os inputs para criação das pranchas (realizada pelo node “*Sheet.ByNameNumberTitleBlockAndViews*”): seus números, seus nomes, as vistas (*viewsports*) e seus respectivos tamanhos. O número é, originalmente, um dado numérico, mas que é convertido para texto (dado tipo *string*). Depois, os nomes de cada folha foram extraídos da planilha, para que a nomenclatura inicial (que está na coluna “nome”, do excel) fosse mantida. Vale ressaltar que os formatos de prancha (A0, A1 etc..) são verificados conforme os tipos de família de folhas já disponíveis no *Revit*. Por último, as vistas correspondentes são vinculadas por meio de um sistema de correspondência entre os nomes listados na planilha e as vistas existentes no projeto, para que cada prancha exiba as plantas, cortes e elevações corretas.

Figura 25: Rotina no dynamo



Fonte: A autora

Figura 26: Criação das pranchas na plataforma



Fonte: A autora

Essa rotina permite gerar todas as pranchas de forma automatizada e consistente, sem a necessidade de ajustes manuais individuais. O método mostra-se particularmente eficiente em projetos com grande volume de pranchas, nos quais o processo tradicional, mesmo em plataformas BIM, seria mais demorado e propenso a erros. Além disso, a solução desenvolvida possui flexibilidade para expansões futuras, como a inclusão automatizada de legendas, carimbos e outros elementos complementares, otimizando ainda mais o fluxo de documentação técnica.

3.3 Gestão de metadados: verificação da área de ventilação mínima de ambientes

A terceira rotina de automação desenvolvida neste estudo é a verificação automática das áreas de ventilação natural. Essa rotina toma como base os critérios estabelecidos na NBR 15.575-4:2021, anexo B (informativo), que orienta sobre ventilação e iluminação naturais. Embora o anexo seja classificado como "informativo" (não obrigatório), ele reflete boas práticas que podem ser adotadas em projetos.

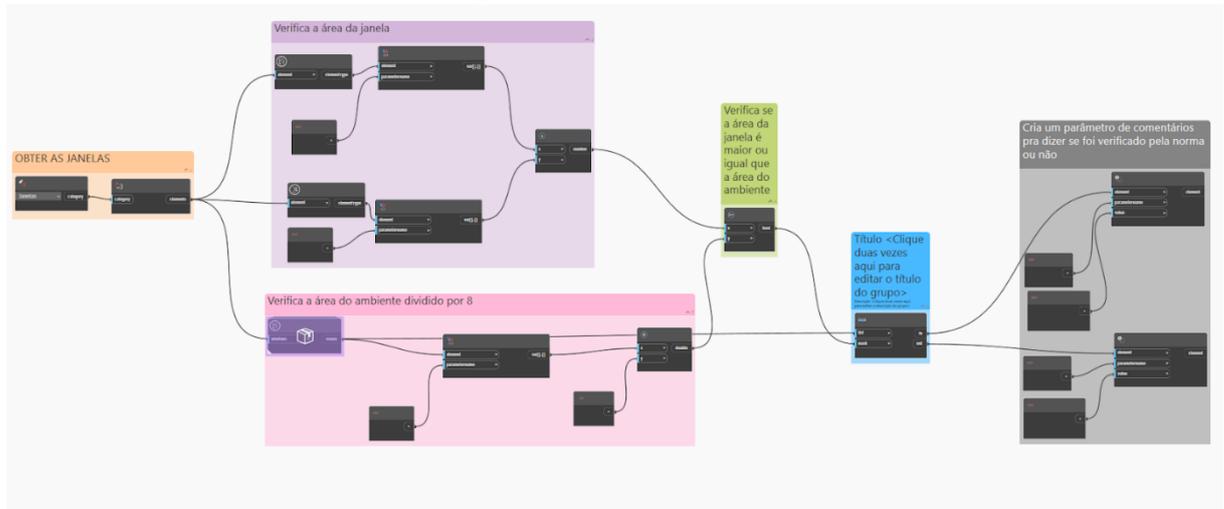
O anexo menciona que:

- Em dormitórios, a área de aberturas para ventilação não deve ser inferior a $1/6$ da área do piso.
- Para outros ambientes (como salas, cozinhas e banheiros), a área mínima pode ser reduzida para $1/8$ da área do piso, desde que atendidas condições específicas de insolação e ventilação cruzada.

Sendo assim, para a aplicação prática desse estudo, considerou-se que a área mínima adotada de ventilação dos ambientes seria $1/8$, e a rotina seguiu cinco etapas lógicas, demonstradas na figura 25:

1. Todas as janelas do modelo são selecionadas. (Grupo laranja)
2. As áreas dessas janelas são calculadas, multiplicando sua largura e altura. (Grupo roxo)
3. Os ambientes que essas janelas estão inseridas são identificados e a área do piso é calculada e dividida por 8 - critério adotado pela norma neste estudo. (Grupo rosa)
4. É criada uma lista com as janelas que atendem e não atendem a norma. (Grupo verde e azul)
5. É criado um parâmetro de comentário para o *Revit* para importação dessa listagem em formato de planilha. (Grupo cinza)

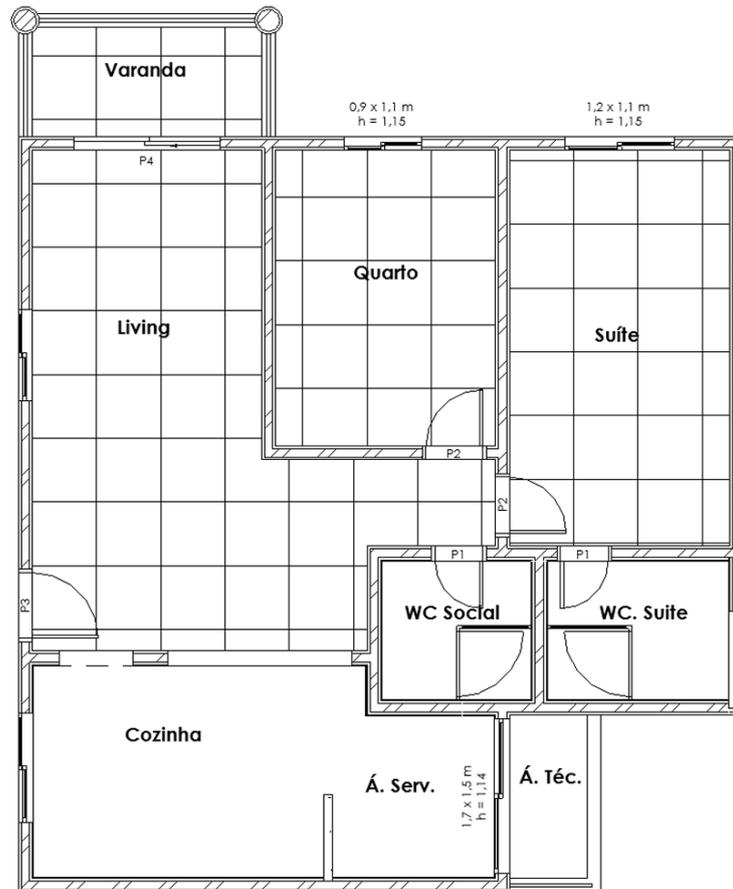
Figura 27: Rotina do estudo



Fonte: A autora

Essa rotina foi testada em um projeto residencial (figura 26).

Figura 28: Projeto residencial adotado como estudo de caso para testagem da rotina

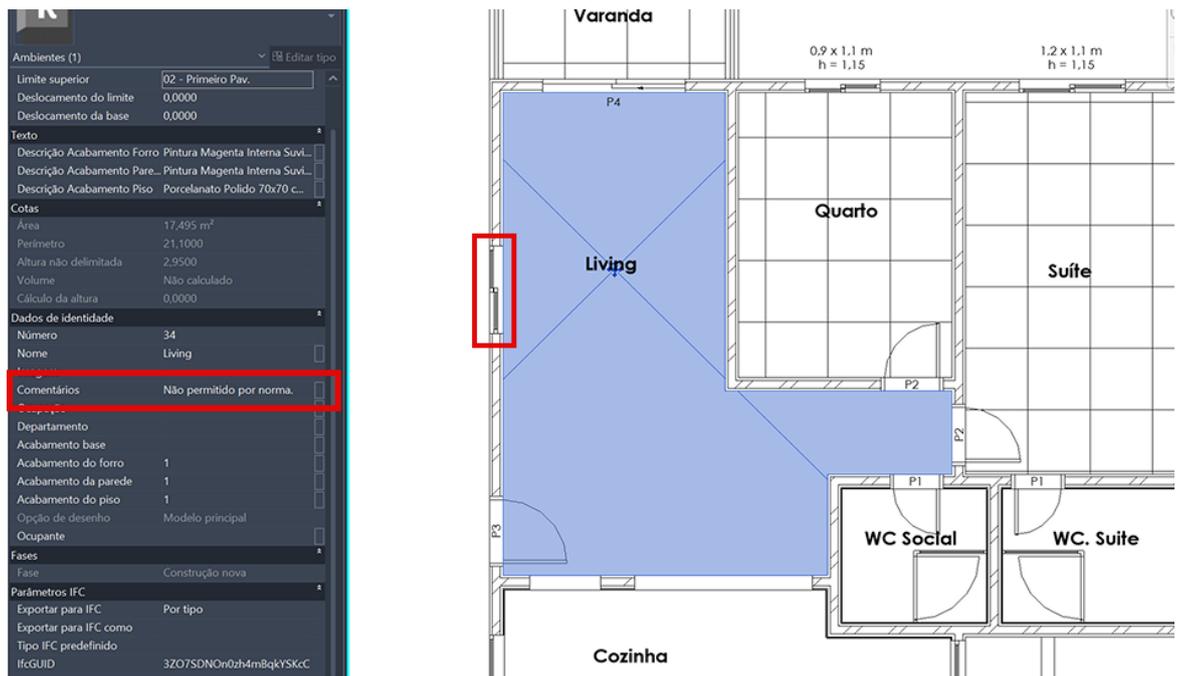


Fonte: A autora, modificado de Caetano (2025)

A identificação dos ambientes classifica-os entre dormitórios e outros cômodos. Para tal classificação é utilizado como filtro o nome (*string*) do ambiente, ou seja, todos os ambientes cujo *string* é dormitório fica em uma lista, enquanto os demais, ficam em outra lista. Para cada janela do modelo, a rotina calcula a área de ventilação. Esses valores são então relacionados às áreas dos ambientes correspondentes.

Como comentado, a verificação final compara a área de ventilação de cada cômodo com a área do ambiente dividida por 8. Os resultados são organizados em uma lista com dado *booleano*¹⁹, que separa os ambientes não conformes dos conformes. O exemplo da figura 27, mostra uma janela que está com sua área de ventilação menor que a área do piso, então seu ambiente correspondente – nesse caso, a sala, é classificado como “não permitido por norma” em seu parâmetro de comentário.

Figura 29: Exemplo de janela que não atendeu à especificação da norma



Fonte: A autora

É importante ressaltar que o parâmetro de comentário foi atribuído a todos os ambientes analisados pela rotina, sendo preenchido automaticamente com:

- "OK" para os ambientes que atendiam aos requisitos normativos

¹⁹ Dado booleano: Tipo de variável que representa valores binários, sendo *True* (verdadeiro) ou *False* (falso), positivo ou negativo, zero ou um, etc.

- "Não permitido por norma" para os casos que apresentaram não conformidade

Essa abordagem apresenta três benefícios fundamentais para o processo de modelagem de projeto. Primeiro, ela proporciona um panorama de todos os ambientes analisados. Segundo, agiliza a detecção dos elementos que precisam de modificações no desenho. Por fim, gera um registro técnico para fins de auditoria e garantia de qualidade.

É importante ressaltar que, para que a rotina funcione adequadamente, é fundamental que o modelo seja desenvolvido seguindo informações padronizadas. A eficiência da rotina depende da consistência nos dados - por exemplo, se o *string* da rotina estiver "dormitórios", todos os ambientes desse tipo precisam estar nomeados exatamente dessa forma, sem variações como "quarto" ou "Dormitório" com letra maiúscula, a menos que a rotina inclua tratamento específico para reconhecer essas variações. Essa padronização representa ao mesmo tempo uma vantagem e um desafio. Por um lado, pode promover mais padronização da modelagem BIM, pois é necessário que cada projetista alimente seu modelo com informações consistentes e seguindo padrões, resultando em projetos mais organizados; por outro, requer treinamento da equipe e verificação para garantir que todos os profissionais envolvidos sigam as mesmas convenções.

A rotina se mostra especialmente útil em processos de aprovação em prefeituras, onde é necessário verificar o atendimento a diversas normas em diversos projetos simultaneamente. Contudo, sua implementação prática exige um esforço inicial para adequar os modelos aos padrões. Superada essa etapa inicial, o processo pode se tornar altamente eficiente, com a execução de rotinas nesse sentido em novos projetos e a exportação dos resultados tornando-se rápidas e sistemáticas.

Vale destacar que a mesma lógica pode ser adaptada para verificar outros requisitos normativos, como acessibilidade ou iluminação, o que amplia consideravelmente sua aplicabilidade em escritórios de arquitetura e órgãos fiscalizadores.

CONSIDERAÇÕES FINAIS

O domínio da modelagem paramétrica revela-se como um potente impulsionador para a prática profissional no campo da AECO. Ao compreender profundamente os princípios da parametrização e da programação visual, o profissional pode desenvolver rotinas que transcendem a fase de modelagem de projetos em plataformas BIM, transformando-se em ferramentas estratégicas para análise e otimização de soluções projetuais.

A integração da programação visual nas etapas de modelagem, documentação e extração de dados permitiu testar diversas variantes projetuais, se mostrando como potencial ferramenta de automatização. Na criação de objetos BIM, observou-se que escritórios ou empresas que trabalham com pré-definições, como em reformas onde a estrutura existente condiciona o projeto, podem tirar maior proveito dessas rotinas, que vão além da modelagem, ao ajudar a testar soluções rapidamente.

Quanto à automação de documentação, os resultados mostram como tarefas cotidianas podem ser simplificadas. As rotinas de criação automática de níveis e pranchas, por exemplo, foram desenvolvidas como demonstração, mas a mesma lógica pode ser aplicada a outras atividades repetitivas na fase de documentação. O mesmo vale para a verificação de normas a partir dos metadados. Uma vez implementada, a rotina se adapta a diferentes contextos, reduzindo erros e retrabalhos.

Entende-se que após esse esforço inicial de configuração, o ganho em produtividade é significativo. Mas o mais relevante é que essa automação não se limita a acelerar processos, ela pode mudar a forma de trabalhar. Ao delegar essas tarefas mecânicas às ferramentas mostradas, os projetistas ganham mais tempo para o que realmente importa: análise, criatividade e tomada de decisão.

Como sugestão para futuras pesquisas, propõe-se a expansão das rotinas paramétricas para outras disciplinas do projeto, como instalações e estrutura, e integração com o *Python* para um refinamento ainda maior. Além disso, é importante pensar na integração com ferramentas de Inteligência Artificial (IA). A combinação da lógica paramétrica com algoritmos de *machine learning* e IA generativa pode potencializar ainda mais a automatização de processos, a exploração de alternativas projetuais, e até a geração de *insights* que podem ajudar no raciocínio das ideias. Paralelamente, também é válido pensar no desenvolvimento de bibliotecas

compartilháveis dessas rotinas, aproveitando que o *Dynamo* é um ambiente de código aberto, que como comentado, permite qualquer usuário adaptar e estender as funcionalidades conforme suas necessidades. Essas contribuições podem ser compartilhadas em fóruns, tutoriais *online* ou até serviços de consultoria técnica, criando um ecossistema colaborativo para essas ferramentas tomarem mais espaço.

Como contribuição, este trabalho explora um repertório de aplicações práticas que podem ser adaptadas a diversos contextos projetuais, servindo tanto como referência técnica quanto como estímulo para a contínua evolução desses métodos na indústria AECO.

REFERÊNCIAS BIBLIOGRÁFICAS

BERNSTEIN, Philip. **Architecture, design, data: Practice competency in the era of computation**. Berlin: De Gruyter, 2018.

BLIKSTEIN, Paulo. **O pensamento computacional e os novos paradigmas da educação**. [S. l.], 2008.

BRASIL. **Decreto nº 10.306, de 2 de abril de 2020**. Institui a Estratégia Nacional de Disseminação do Building Information Modeling. Diário Oficial da União: seção 1, Brasília, DF, p. 1, 3 abr. 2020. Disponível em: < http://www.planalto.gov.br/ccivil_03/_ato2019-2022/2020/decreto/D10306.htm > Acesso em: 17 mai. 2025.

BRENNAN, Karen; RESNICK, Mitchel. **New frameworks for studying and assessing the development of computational thinking**. In: ANNUAL MEETING OF THE AMERICAN EDUCATIONAL RESEARCH ASSOCIATION, 2012, Vancouver. Proceedings... Vancouver: AERA, 2012. Disponível em: < <https://scratched.gse.harvard.edu/ct/files/AERA2012.pdf> > Acesso em: 10 jul. 2024.

CYPHER, Allen et al. **Watch What I Do: Programming by Demonstration**. Cambridge: MIT Press, 2010.

EASTMAN, C. et al. **BIM Handbook: a guide to building information modeling for owners, managers, designers, engineers, and contractors**. 2. ed. Hoboken: Wiley, 2011.

EASTMAN, Charles et al. **An Outline of the Building Description System**. Pittsburgh: Institute of Physical Planning, Carnegie-Mellon University, 1974.

FLÓRIO, Wilson. **Modelagem paramétrica e fabricação digital na arquitetura: reflexões sobre o ensino de projeto**. In: ENCONTRO DA ASSOCIAÇÃO NACIONAL DE PESQUISA E PÓS-GRADUAÇÃO EM ARQUITETURA E URBANISMO, 3., 2014, São Paulo. Anais eletrônicos. São Paulo: ANPARQ, 2014.

Disponível em < <https://www.anparq.org.br/dvd-enanparq-3/htm/Artigos/ST/ST-NPNT-008-05-FLORIO.pdf> > Acesso em: 21 jun. 2025.

GIEDION, Sigfried. **Espaço, tempo e arquitetura: o desenvolvimento de uma nova tradição**. 5. ed. São Paulo: Martins Fontes, 2008.

GREEN, Tristan. **Visual Programming Languages: A Historical Perspective**. Journal of Computational, 2021.

GRIZ, Cristiana; ANDRADE, Max L.; RESENDE, Camila. **Parametric modeling in the initial design of an architectural project: a didactic experience**. SIGraDi 2023. Disponível em: < <https://pdf.blucher.com.br/designproceedings/sigradi2023/163.pdf> > Acesso em: 10 mar. 2025.

INGRAM, J. **Understanding BIM: the past, present and future**. Abingdon: Routledge, 2020.

LEACH, Neil. **Parametrics Explained**. 2014. Disponível em: < <https://pt.scribd.com/document/803423422/Parametrics-Explained> > Acesso em: 21 jun. 2025.

MYERS, Brad A. **Natural programming languages and environments**. Communications of the ACM, v. 49, n. 3, p. 47-52, mar. 2006. Disponível em: < <https://dl.acm.org/doi/pdf/10.1145/1015864.1015888>. > Acesso em: 10 mai. 2025.

POLONINI, Flavia. **Modelagem paramétrica como ferramenta de projeto: uma abordagem metodológica**. 2014. 215 f. Dissertação (Mestrado em Arquitetura e Urbanismo) – Faculdade de Arquitetura, Universidade Federal da Bahia, Salvador, 2014. Disponível em: < <https://repositorio.ufba.br/handle/ri/15339> > Acesso em: 21 jun. 2025.

SILVA, João Carlos. **Da prática à representação: a evolução do conhecimento construtivo na arquitetura**. São Paulo: Editora da Universidade de São Paulo, 2008.

SILVA, João M. **Lógica computacional aplicada: fundamentos de programação para arquitetura e design**. Porto Alegre: Bookman, 2020.

SUCCAR, Bassam. **Building Information Modelling Framework: A Research and Delivery Foundation for Industry Stakeholders**. Automation in Construction, v. 18, n. 3, p. 357-375, 2009. Disponível em: < <https://doi.org/10.1016/j.autcon.2008.10.003> > Acesso em: 7 jun. 2025.

TEDESCHI, Artur. **AAD - Algorithms-Aided Design: Parametric Strategies using Grasshopper**. Potenza: Le Penseur, 2014.

WOODBURY, Robert. **Elements of parametric design**. New York: Routledge, 2010.