



SIGRADI2018
TECHNOPOLITICAS
xxii congresso da sociedade
iberoamericana de gráfica digital
22th conference of the
iberoamerican society
of digital graphics
07|08|09|novembro|2018
iau usp | são carlos | sp br

Designers of the XXI century: BIM software programming and the development of new competencies

Juliano Lima da Silva

Faculdade Meridional IMED | Brazil | juliano_lima_silva@hotmail.com

Andréa Quadrado Mussi

Faculdade Meridional IMED | Brazil | andrea.mussi@imed.edu.br

Tháisa Leal da Silva

Faculdade Meridional IMED | Brazil | thaisa.silva@imed.edu.br

Paola Zardo

Faculdade Meridional IMED | Brazil | pazardo@gmail.com

Abstract

This paper presents the scenario of programming use by architects and engineers by creation of their own unique tools. It aims to emulate and understand BIM software customization phenomenon by development of plug-ins. Design Science Research was used to direct the construction of artifacts for specific practical problems. Results are presented by development of Revit plug-ins to meet Brazilian Performance Standard criteria in areas of acoustic and luminous performance. It is concluded that scripting enables the designer to tailor processes to their own particularities, possibly becoming a high-impact skill in the future, with importance regarding their independence and versatility.

Keywords: BIM; Programming; Design science research; Dynamo; Performance Standard.

1

INTRODUÇÃO

Este trabalho visa identificar e caracterizar o processo de criação de ferramentas pelos profissionais por meio de desenvolvimento de plug-ins em software BIM. Primeiramente é descrito o cenário de inovações e novas competências na arquitetura e engenharia, então são apresentados os objetivos da pesquisa e sua delimitação.

Dentre as tecnologias digitais, destacam-se os softwares CAD (Computer Aided Design), introduzidos nos escritórios de arquitetura no final do século 20, fazendo com que as atividades de desenho em papel aos poucos fossem substituídas pelos softwares, os quais proporcionavam maior produtividade nos processos de trabalho. Embora isto tenha causado uma redução no número de profissionais desenhistas, tal fato possibilitou ao campo de arquitetura uma nova interface – a inclusão de conceitos de tecnologia da informação, o que amplificou o potencial de complexidades no ambiente projetual (Celani et al., 2015).

Com este avanço das tecnologias no ambiente de projeto, houve também o surgimento de novas competências para os profissionais de arquitetura e engenharia civil, possibilitando a utilização de diversas ferramentas computacionais para dar suporte ao desenvolvimento de projetos e visando o aumento da confiabilidade das informações de uma construção. Tais competências trouxeram mérito para estas profissões e se tornaram mais aparentes conforme os níveis de informações dos

projetos aumentaram (Garber, 2009). Celani et al. (2015) ressaltam que uma mudança semelhante à do CAD em relação ao desenho manual pode ocorrer conforme as ferramentas necessitarem de novas competências.

Com o uso de novas tecnologias e processos como o BIM (*Building Information Model*), em um cenário cada vez mais competitivo, o setor AEC (Arquitetura, Engenharia e Construção) está requisitando uma quantidade cada vez maior de informações relativas às soluções de custo, performance e gerenciamento, que remetem à interdisciplinaridade projetual. Os projetistas precisam abraçar as novas abordagens, ou correm o risco de terem seu papel severamente diminuído ou substituído por profissionais com as novas competências exigidas pelo mercado (Holzer, 2015).

Além disso, conforme o número de soluções inovadoras aumenta, também aumenta a complexidade de associar decisões de projeto aos novos aspectos introduzidos, o que torna o processo de tomada de decisão sobrecarregado de informações de difícil gerenciamento, tornando métodos simplificados ou manuais algo cada vez mais complicado a um nível compreensível. Como profissionais, arquitetos e engenheiros agregam valor aos projetos por suas decisões e, portanto, devem realizá-las enquanto interpretam a vasta informação disponível, aplicando-a de maneira eficiente (PITTMAN, 2003).

Esta necessidade de melhorar o gerenciamento das informações de diversas naturezas em projetos foi um dos motivos que levou à ascensão do BIM, modificando radicalmente o paradigma no qual os arquitetos e engenheiros estavam inseridos na medida em que seus processos de projeto se tornaram mais colaborativos, integrados e interdisciplinares (Garber, 2009). Atividades não rotineiras também podem ser automatizadas até certo ponto, se a informação disponível ao projetista for suficiente (Celani et al., 2015).

Imagina-se um cenário onde todos os projetistas podem compartilhar as informações de suas instâncias de projeto em um mesmo ambiente. Um modelo tridimensional dotado de informações precisas, inteligentes e interconectadas entre as disciplinas, onde os profissionais podem ter acesso a informações relevantes a seu projeto no momento em que precisam, de modo que trabalhem de forma coordenada. Tal cenário é alcançável por meio da implementação do BIM nos processos de projeto.

Pois, enquanto as informações acionáveis de um desenho feito em software CAD 2D dependem exclusivamente da interpretação dos projetos, em modelos BIM é possível incluir informações semânticas ricas em dados, que vão além das dimensões de formas arquitetônicas, para agregar conhecimento acerca de performance, função e custo – aspectos que compõe uma edificação real (Pittman, 2003).

Neste cenário, alguns autores vêm repensando o papel dos projetistas, tanto na maneira como projetos são pensados e desenvolvidos (Holzer, 2015), como no escopo de tomada de decisão (Keough; Hauck, 2017) e na intensificação do uso das ferramentas para otimização e design generativo (Garber, 2009).

Conforme Pittman (2003), o desafio do projetista futuro é integrar BIM com o desenvolvimento de ferramentas que possibilitem o consumo de informações de forma interoperável e digitalmente – em detrimento da impressão em papel ou mera geração de documentos, são necessários mecanismos que possibilitem a análise dos aspectos de performance, função e custo de seus projetos – ferramentas integradas inseridas no paradigma de projeto do BIM.

Uma grande preocupação de projetistas é de que os computadores irão substituir suas habilidades ou torná-las obsoletas, contudo, a computação na arquitetura ainda possui um grande foco no aspecto da produtividade de geração de documentação e gerenciamento de informações – atividades que não envolvem, ou pouco influenciam na criatividade artística dos profissionais (Celani et al., 2015).

Os autores reforçam que, as etapas de cálculo, simulações térmicas, lumínicas e acústicas, verificações de desempenho, normatização (*codechecking*) são automatizáveis, pois envolvem características de componentes e materiais, relações semânticas entre tais componentes e, principalmente, regras ou etapas bem definidas, diferentemente da concepção, que dificilmente é programável.

Diante destas situações, Gauchat (2009) defende que, embora muitos pensem que o projetista que utiliza CAD 2D será substituído pelo projetista que utiliza BIM, a crescente complexidade de projetos e a necessidade de inovar, projetar eficientemente e economizar no setor AEC tornará o projetista BIM um profissional que precisa de sólida interdisciplinaridade, habilidade e poder para tomar decisões.

Por ampliar a possibilidade de agregar valor à construção, o BIM pode modificar radicalmente a profissão dos arquitetos e engenheiros na medida em que aumenta sua responsabilidade e poder de desenvolver soluções inovadoras no ambiente de projeto (Gauchat, 2009). Tais responsabilidades adquiridas colocam o arquiteto ou engenheiro como um profissional novamente no centro dos processos, pois toda informação acerca da edificação é contida nos modelos de informação (Garber, 2009).

Este cenário tem mais impacto no design, construtibilidade e custo das edificações (Pittman, 2003). Portanto, com uma importância muitas vezes maior que a tradicionalmente percebida – do projetista que apenas desenha, elevando as profissões para um novo patamar – do projetista que ativamente soluciona e toma decisões significativas, reiteradas pela abundante quantidade de informação e ferramentas sofisticadas de simulação.

Em relação às competências da era digital no âmbito do desenvolvimento de soluções inovadoras, Holzer (2015) descreve o *Scripting*, como a capacidade de projetistas utilizarem a programação para customizar seus processos e escolhas de projeto.

Contudo, há uma grande lacuna entre o projetista que simplesmente sabe utilizar ferramentas e softwares comerciais – necessitando apenas de treinamento, e o projetista que possui conhecimento interdisciplinar entre arquitetura e computação, ou tecnologia de informação (Celani et al., 2015).

Além disso, a programação em softwares de arquitetura por vezes não é realizada por arquitetos e sim por profissionais exclusivamente de áreas da computação ou afins. Aish (2003) acredita em uma lacuna neste contexto e questiona: “existe um encontro entre as mentes dos projetistas criativos que utilizam os sistemas CAD e os engenheiros de software que os desenvolveram?” (traduzido de Aish, 2003, p. 338).

Para responder esta pergunta, em outra pesquisa, Aish (2013) relata que a história do CAD está atualmente passando por algo chamado de era *Design Computation*, onde se busca ultrapassar limitações de plataformas e ampliar potenciais de CAD e principalmente BIM a partir do aproveitamento da programação em projetos – que se manifesta por meio das formas complexas, otimização, gerenciamento de dados, design generativo e evolutivo, e integração com a fabricação, por ferramentas desenvolvidas pelos próprios projetistas.

Nesta nova realidade, “o papel do arquiteto deve ser reconceituado daquele de um gênio romântico ou um guru tecnológico, para um estrategista multidisciplinar” (traduzido de Ottchen, 2009, p.25).

Contudo, mais do que isto, softwares de arquitetura devem possibilitar o processo exploratório de design – a criação e manipulação de relações geométricas e a interpretação dos impactos decorrentes, facilitando o processo de tomada de decisão, ao passo em que amplificam as habilidades intelectuais do usuário (Aish, 2003). Para o cenário de projetos, inovações na automatização e integração de projetos não significam que os projetos serão totalmente automatizados, mas sim que possibilitarão aos projetistas dedicar mais tempo e ter mais liberdade para atuar nas decisões importantes – soluções de projeto, em detrimento das atividades operacionais e de baixo impacto.

Em relação às demandas de programação de projetistas, Aish (2003) relata que muitos dos softwares CAD possuem interfaces focadas apenas na geração de documentação gráfica, não atendendo a exploração da criação e customização de maneira integrada. O autor afirma que recursos para tal exploração existem escondidos nas linhas de comando e funções internas dos softwares, “para um projetista com habilidades de programação, isto proporciona um recurso extremamente poderoso” (traduzido de Aish, 2003, p. 340). Para Garber (2009), muitas das características generativas e scripts de programação dos softwares BIM são despercebidos ou não utilizados por muitos projetistas, sendo necessário trazer o foco do processo de projeto para a exploração de soluções alternativas ao invés da mera representação gráfica ou documental das escolhas (Aish, 2013).

Aish (2003) destaca que a competência de programar possibilita, portanto, atingir novos patamares de compreensão das ferramentas. Isto permite que problemas específicos e particularidades de projeto sejam contempladas por meio de soluções customizadas e direcionadas (Ferreira; Leitão, 2015), programadas pelos arquitetos e engenheiros, e não apenas por especialistas.

O desafio é encorajar praticantes de arquitetura, engenharia e design com pouco conhecimento acerca de programação a programar, visto que este novo paradigma é encarado como uma barreira de abstração, ainda mais intenso nas TPL (*Textual Programming Languages*) (AISH, 2013b). Uma alternativa para tal está nos softwares de VPL (*Visual Programming Languages*), como o Dynamo, extensão que opera sobre a plataforma de modelagem de arquitetura Autodesk Revit, por meio de workflows visuais de feedback imediato (Dynamo, 2018).

O Dynamo opera sob o *DesignScript*, linguagem de programação que permite que projetistas, em diversos níveis de conhecimento acerca de programação utilizem componentes para criar seus próprios programas para manipular modelos BIM (AISH et al., 2012). Isto torna o projetista ainda mais independente na medida em que possibilita que resolva muitos de seus entraves e particularidades por conta própria, não necessitando esperar que os grandes desenvolvedores criem funcionalidades específicas na interface das plataformas.

A intenção por trás da criação do *DesignScript* foi ultrapassar as barreiras de aprendizado de projetistas novatos em programação por meio da integração entre programação visual e textual (Aish; Mendoza, 2016). Os

autores ressaltam que as programações visual e textual podem se complementar, em variados graus de intensidade e complexidade, ao longo do aprendizado do projetista em relação à criação de suas soluções. Conforme as rotinas desenvolvidas se tornam mais complexas, Leitão, Santos e Lopes (2012) defendem o uso de linguagens de programação textual concomitantemente com visual através dos componentes híbridos.

A divisão de funções em componentes ou nodos, presente nas extensões as transforma em verdadeiras “caixas de ferramentas” (Mackey; Roudsari, 2018), permitindo que as soluções criadas por usuários sejam altamente customizáveis e compartilháveis. Muitas das extensões VPL são softwares de código aberto e, portanto, encorajam tal comportamento. Além disso, estas situações fomentam a criação das comunidades de projetistas programadores, fóruns virtuais onde os membros discutem e produzem suas soluções customizadas, auxiliando uns aos outros no processo de aprendizado e criação de rotinas de programação, por vezes disponibilizando-as para que outros testem e avaliem seus códigos.

Assim, a partir do conhecimento acerca do cenário de inovações tecnológicas, e do desenvolvimento de ferramentas por meio da competência de programar na arquitetura e engenharia, este trabalho propõe compreender o fenômeno de criação de ferramentas pelos projetistas, por meio de uma experiência de construção de programas pelo pesquisador.

Para isto, na pesquisa desenvolvida utilizou-se o método *Design Science Research*, que possui uma abordagem de caráter prescritivo, direcionando o desenvolvimento de artefatos inovadores voltados à resolução de um problema específico, de modo a proporcionar vínculo com conhecimentos teóricos em relação a um cenário ou organização, e possibilitando a geração de discussões e reflexões sobre a experiência (Dresch; Lacerda; Antunes Jr., 2015).

Para emular a situação do projetista programador, foi necessário escolher um tema específico para desenvolvimento dos artefatos, para tal, definiu-se o atendimento de requisitos da Norma de Desempenho Brasileira (ABNT, 2003). Os requisitos escolhidos foram o atendimento ao desempenho acústico em relação ao ruído aéreo de vedações verticais, e o atendimento aos níveis de iluminação natural no interior de ambientes de edificações habitacionais.

O desenvolvimento dos artefatos foi realizado por meio de plug-ins para o software Autodesk Revit, como uma maneira de integrar as ferramentas no processo BIM. A escolha de programar plug-ins voltados para a Norma de Desempenho permitiu a compreensão acerca do processo de criação de ferramentas diretamente pelo pesquisador, possibilitando explorar problemas específicos de projeto por meio de *codechecking* e avaliar como tal abordagem poderia ser viável em outras situações. Assim, tornou-se possível estruturar um arcabouço teórico sobre o cenário apresentado e, portanto, uma reflexão e contribuição científica alinhada com a metodologia.

Nas seções a seguir, serão apresentados os procedimentos e definições para a construção dos artefatos visando o atendimento dos requisitos específicos da Norma de Desempenho Brasileira; os resultados alcançados a partir do desenvolvimento desses artefatos; e finalmente as conclusões acerca do trabalho desenvolvido.

METODOLOGIA

Esta seção detalha os procedimentos de desenvolvimento dos artefatos, uma das etapas da pesquisa inserida no método *Design Science Research*, de caráter essencialmente criativo (Dresch; Lacerda; Antunes Jr., 2015). Os plug-ins propostos se enquadram no método, na categoria de instanciação, pois caracterizam-se na realização do artefato em seu ambiente, por meio da operacionalização de ferramentas, ou ainda, por meio de uma articulação do cenário inicialmente apresentado (March; Smith, 1995).

Em relação ao cenário de problemas oriundos dos processos de projeto tradicionais e da falta de adequação de componentes BIM a normativas brasileiras, destaca-se a lacuna de plug-ins voltados para análises de performance das edificações, bem como a desconectividade de tais informações a dados de referência para projetistas, sendo estas algumas das motivações para o desenvolvimento dos artefatos direcionados para o tema específico escolhido.

Assim, neste trabalho buscou-se identificar alguns problemas característicos do processo de projeto, relacionados ao atendimento à Norma de Desempenho. Para isso, alternativas baseadas na programação de funcionalidades em plataformas BIM foram levantadas, entre elas, o desenvolvimento de plug-ins, de modo a possibilitar a qualificação dos processos de projeto voltados para o desempenho (Figura 1).

A plataforma Autodesk Revit foi escolhida como o software BIM a ser programado devido à sua comunidade de usuários já estabelecida e suas possibilidades de customização (Autodesk, 2017). Além disso, como muitas das avaliações de desempenho precisam ser comprovadas por meio de projetos documentados, a iniciativa desta etapa metodológica parte da premissa de que informações dos modelos BIM possam ser manipuladas por programação para verificar, calcular ou simular o desempenho de sistemas construtivos em uma edificação, diretamente do ambiente BIM pelo Revit.

Dynamo foi escolhido como a maneira de desenvolver os plug-ins, pois possibilita a manipulação de dados do modelo de arquitetura, bem como a automatização de processos onde haja a necessidade de interações e repetição, que utilize parâmetros compartilhados ou que envolvam design paramétrico de modo geral.

Contudo, foi necessário definir critérios de problemas de projeto que utilizem tais potenciais de programação. Com isso, situações de projeto que facilmente podem ser resolvidas no próprio Revit, por cálculos simples, por planilhas no Excel ou que já possuam um software próprio foram descartadas. Foram escolhidas situações onde os plug-ins realizem tarefas que dificilmente possam ser realizadas utilizando apenas funções básicas do Revit, que utilizem a complexidade de dados de famílias e encadeamento lógico, ou ainda, que proporcionem funcionalidades interativas com o usuário, que possibilitem auxiliar nos processos de escolha de soluções tecnológicas de sistemas construtivos enquanto o projetista realiza a modelagem de arquitetura – possibilitando ter feedback acerca do desempenho da edificação já nas etapas iniciais de desenvolvimento do projeto.

Desta maneira, o funcionamento geral (Figura 2) desejado baseia-se no uso da modelagem de arquitetura para extração de informações geométricas e parâmetros, dados que são manipulados internamente por meio da extensão Dynamo e seus pacotes de funções (*Dynamo Packages*), na forma de rotinas de programação. Então, após o processamento, informações relativas a cálculos, verificações e simulações, relacionadas às estimativas de desempenho da edificação, são apresentadas ao usuário, por meio de listas, tabelas, e filtros de vista de elementos BIM, podendo ser utilizados em vistas ou pranchas do Revit para gerar documentação de desempenho da edificação.

Tal processo precisa ser dado por meio de uma interface de fácil utilização, de modo que projetistas com conhecimento acerca da NBR 15.575 possam usar os plug-ins para alavancar seus processos de projeto. A intermediação entre as rotinas de programação e a inserção de inputs pelo usuário é dada no organizador de rotinas *Dyno Browser* (ProRuBIM, 2018), aliado ao pacote de funções de interface *Data-Shapes*.

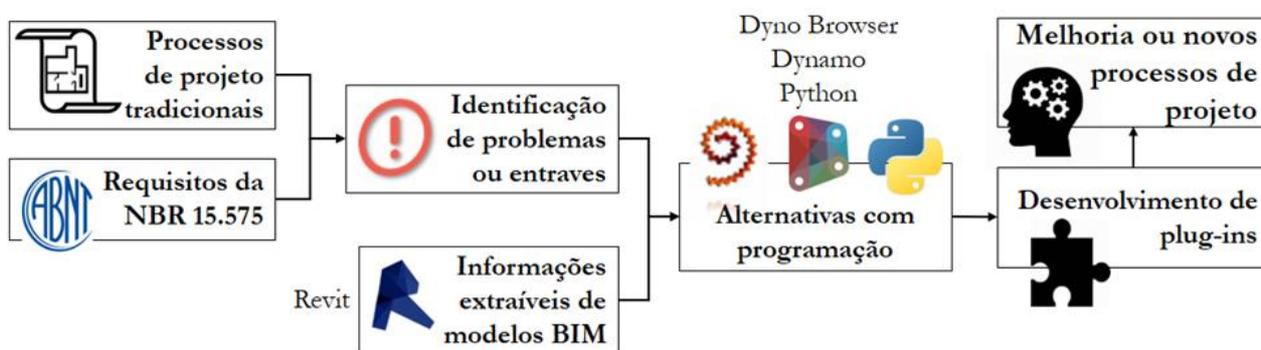


Figura 1: Procedimentos utilizados para o desenvolvimento dos plug-ins (Fonte: Autores).

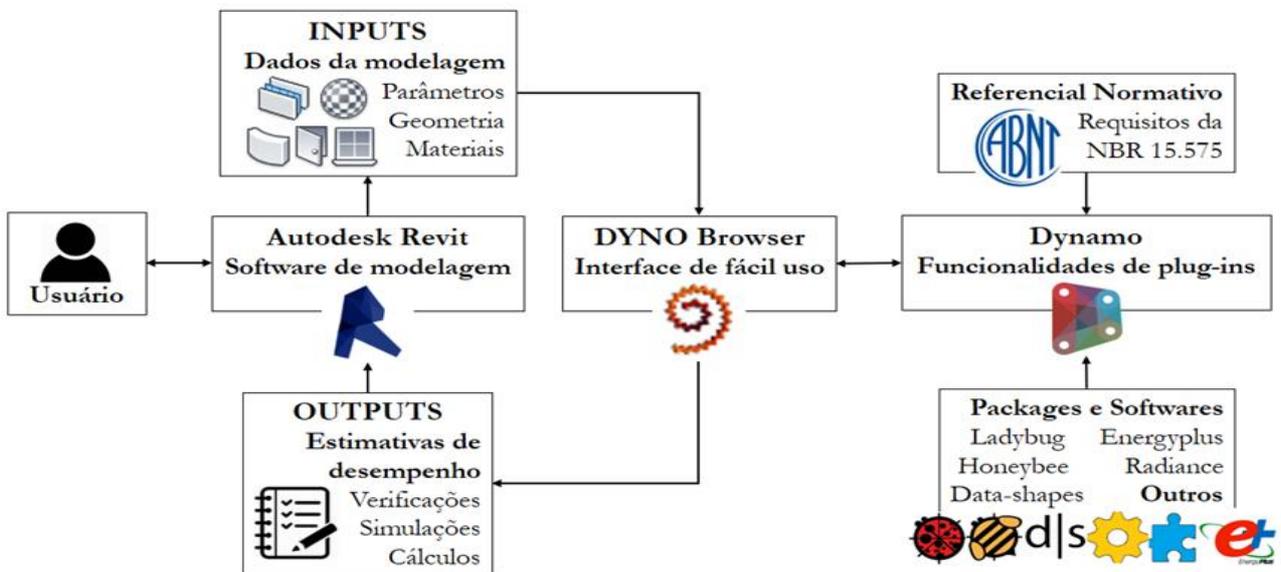


Figura 2: Fluxograma do funcionamento geral dos plug-ins propostos (Fonte: Autores).

A partir desta abordagem, dois plug-ins foram desenvolvidos: um para a estimativa do desempenho acústico de vedações verticais; e outro para a estimativa de níveis de iluminância natural no interior de ambientes. Ambas as ferramentas são ligadas a dados de referência da NBR 15.575 para suas interações. A seguir, são apresentados os resultados do desenvolvimento das ferramentas, por meio da demonstração de seu uso em algumas situações exemplo.

RESULTADOS

Para o plug-in de desempenho acústico, foi implementado um algoritmo no Dynamo para realizar o cálculo do índice de redução sonora de vedações verticais, por meio do preenchimento de parâmetros compartilhados, inserindo informações de acordo com bancos de dados de alguns sistemas construtivos ou dados de fabricantes. O algoritmo define o valor do fator RW de portas, janelas e paredes na vista atual de projeto no Revit conforme seus respectivos sistemas construtivos.

Então, a partir dos parâmetros de área de cada instância de parede, janela e porta na vista atual, o algoritmo detecta elementos de parede do tipo hospedeiro (*Host Walls*) e suas respectivas janelas e portas (*Hosted Elements*), calculando o índice de redução sonora ponderado do sistema, identificado como RW sistema. Para demonstração do funcionamento do algoritmo, algumas configurações de parede e esquadrias foram criadas como famílias do Revit (Figura 3).

Verifica-se que durante a estimativa de desempenho acústico o índice de redução sonora das esquadrias – portas e janelas, pode ter um impacto considerável no desempenho do sistema construtivo como um todo, visto que é necessária uma esquadria de boa qualidade para manter o valor ponderado nos níveis da norma.

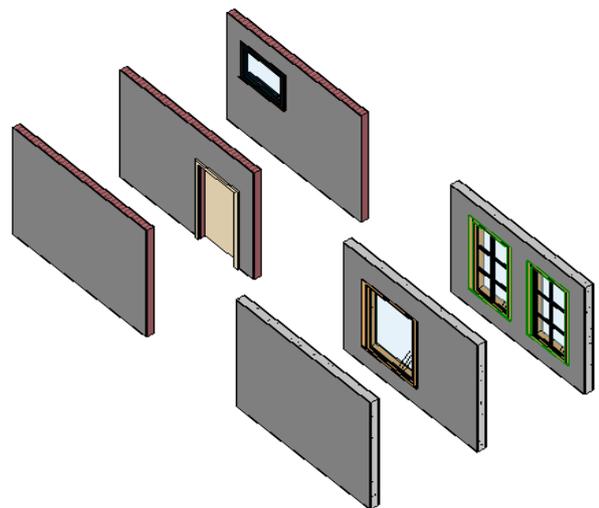


Figura 3: Modelagem de elementos de parede e suas respectivas esquadrias no Revit (Fonte: Autores).

Assim, foram realizadas algumas interações com cenários variados, para exemplificar estas situações, apresentadas na Figura 4. Foram verificados elementos de parede de concreto ($Rw = 45\text{dB}$), nas situações sem esquadria, com uma janela ($Rw = 36\text{dB}$), e com duas janelas ($Rw = 27\text{dB}$ em ambas); então, modelos de parede de alvenaria ($Rw = 40\text{dB}$), nas situações sem esquadria, com uma porta ($Rw = 28\text{dB}$), e com uma janela ($Rw = 26\text{dB}$).

Para melhor visualização dos resultados, os valores dos índices calculados são apresentados como objetos de anotação do Revit, remetendo aos valores do parâmetro de cálculo encontrados em cada conjunto de parede e esquadrias. É possível verificar que os valores dos sistemas construtivos são reduzidos conforme o desempenho da esquadria empregada, o que pode impactar a decisão do projetista.

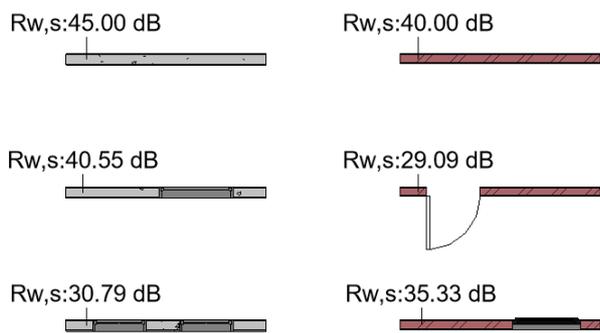


Figura 4: Vista da planta baixa das paredes modeladas e cálculo do índice de redução sonora ponderado (Fonte: Autores).

Um segundo algoritmo do plug-in efetua o preenchimento dos elementos verificados com uma escala de cores customizada. O potencial de uso da função está na possibilidade de auxiliar profissionais da AEC na estimativa da performance acústica de seus projetos ao facilitar a compreensão do impacto de diferentes decisões de materiais de paredes e esquadrias.

Ao executar a função de preenchimento por cores, o usuário pode definir o valor R_w desejado. Desta forma, o algoritmo interpreta este input e cria uma série de filtros de vista na interface do Revit, configurados com condicionantes em uma margem de -5 dB e $+5$ dB do valor estipulado, para alterar a representação de cores de elementos de parede. O objetivo destas margens de valores é emular os critérios de níveis de desempenho da norma: Mínimo, Intermediário, e Superior.

Na Figura 5, as situações verificadas anteriormente são submetidas aos filtros de cores, o que permite visualizar facilmente o atendimento ao valor estipulado de 30dB.

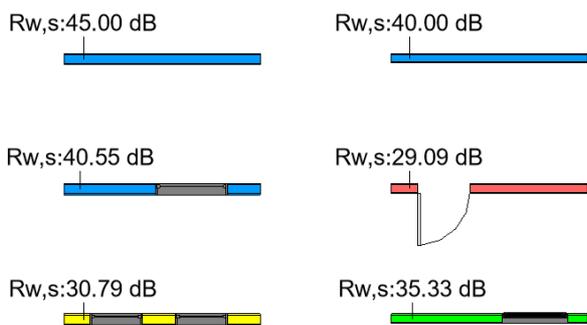


Figura 5: Representação por cores para o valor de desempenho 30dB, onde: vermelho – não atende; amarelo – Mínimo; verde – Intermediário; azul – Superior (Fonte: Autores).

A segunda ferramenta desenvolvida é o plug-in para cálculo de iluminâncias, levando em conta apenas a luz natural no interior de ambientes, conforme os critérios estabelecidos pela Norma de Desempenho Brasileira.

Dentre os diversos pacotes de funções utilizados no ambiente Dynamo, destacaram-se Ladybug e Honeybee, pacotes originários do plug-in Grasshopper para Rhino 3D, mas que recentemente estão sendo adaptados para a plataforma BIM. Ladybug e Honeybee oferecem

soluções voltadas para análises energéticas, que possibilitam o uso de funções para a eficiência e design paramétrico orientado ao desempenho (LADYBUG, 2018). Assim, o plug-in implementado é uma adaptação de um conjunto de ferramentas já existentes, customizadas para serem inseridas no contexto da Norma de Desempenho Brasileira.

Ao executar o plug-in, inicialmente, o usuário é apresentado a uma série de inputs necessários: um arquivo .epw da localização da edificação, dados de dia, mês, horário e tipo de análise (Iluminância ou Fator de Luz Diurna). Os ambientes a serem analisados também podem ser escolhidos e confirmados no Revit por meio de um botão que permite apenas a seleção de elementos BIM do tipo "Room". Esta série de parâmetros então é gravada no algoritmo, e em conjunto com informações de paredes e de aberturas, possibilita a geração de um grid de análise, criando uma série de elementos genéricos numerados dentro de cada ambiente analisado. Os elementos genéricos são superfícies espaçadas a cada 0,5m, posicionadas a uma altura de 0,75m do piso.

A partir da análise do grid, são realizadas operações internamente na programação por meio de componentes do Ladybug que disponibilizam os cálculos por Ray-Tracing pelo software Radiance, capturando as informações geométricas e climáticas e gerando uma lista de valores resultantes de iluminância. Os resultados desta lista de valores são então pareados com cada geometria gerada no grid, assim, cada respectivo elemento genérico possui um valor definido no parâmetro.

Assim como realizado no plug-in para desempenho acústico, os elementos podem ser preenchidos com uma escala de cores. Para exemplificar isto, foi modelado um ambiente no Revit com duas aberturas, configurado para a cidade de Passo Fundo/RS, com as aberturas para o norte, em uma análise às 9h30min. Na Figura 6, é possível visualizar os resultados das análises em parâmetros dos componentes do grid, bem como a escala de cores aplicada, remetendo aos valores encontrados em cada instância.

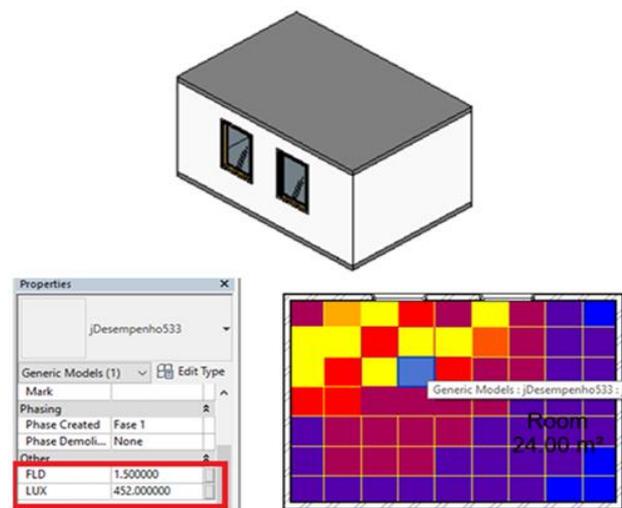


Figura 6: Modelagem de ambiente com aberturas e parâmetros gerados pela análise de iluminâncias internas (Fonte: Autores).

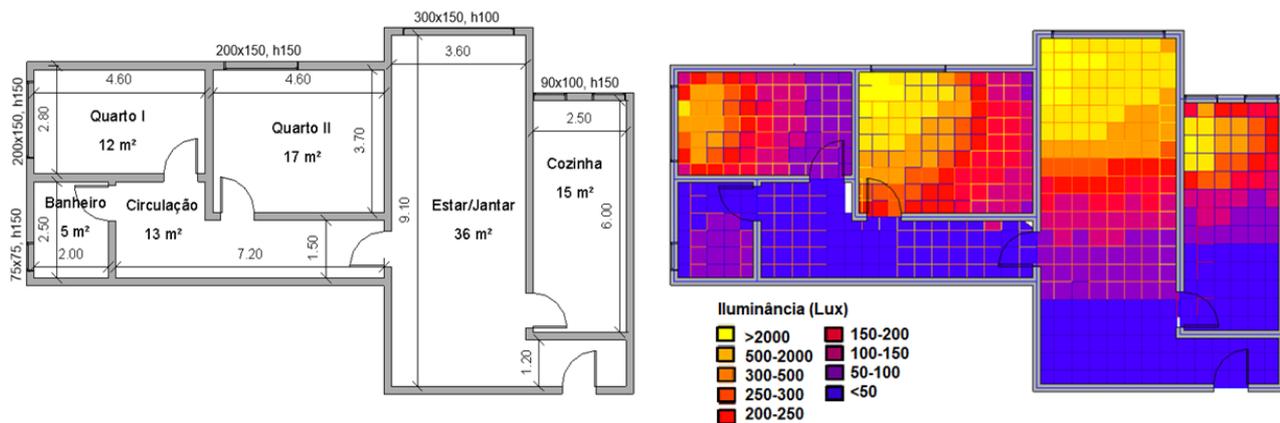


Figura 7: Modelagem de projeto habitacional e iluminância calculada em cada ambiente no horário 09h30min (Fonte: Autores).

Com as funções de geração de grids e preenchimento com escala de cores é possível verificar se certas composições de ambientes e aberturas atendem aos níveis de desempenho estipulados na normativa, bem como para o desenvolvimento de relatórios. Devido ao fato de que a Norma de Desempenho estipula que o período de análise deve ser feito nos dias 23 de abril e 23 de outubro, nas horas 9:30 e 15:30, é possível configurar estas configurações diretamente no plug-in.

Um segundo exemplo de aplicação do plug-in, para um projeto exemplo, é demonstrado na Figura 7, com uma análise de iluminância para todos os ambientes.

DISCUSSÃO

Este trabalho utilizou o método Design Science Research para direcionar o desenvolvimento de plug-ins para um software BIM como uma maneira de compreender o cenário de inovações e novas competências na arquitetura e engenharia.

Para a criação das ferramentas foi utilizada a extensão Dynamo, que requereu do pesquisador conhecimentos em programação, cujo aprendizado e uso foram facilitados pelo emprego de *Designscript*, uma linguagem de programação visual, intuitiva e que permite feedbacks constantes sobre o que foi desenvolvido.

Para exemplificar o cenário do projetista que desenvolve suas próprias soluções, foram apresentados dois plug-ins com temáticas específicas: o atendimento de requisitos de desempenho acústico e lumínico de edificações habitacionais conforme a Norma de Desempenho Brasileira.

A experiência possibilitou a criação de ferramentas de cálculo, verificação e simulação direcionadas à Norma de Desempenho, que dificilmente seriam disponibilizadas ou planejadas pelos desenvolvedores das grandes plataformas. Assim, considera-se o uso da programação como uma maneira de ultrapassar estes limites e possibilitar a customização de soluções, atendendo as mais particulares situações de projeto.

Da mesma maneira como foram desenvolvidas soluções específicas para atender requisitos de uma normativa, poderiam ser desenvolvidos plug-ins direcionados a automatização de outros aspectos de projeto. Alguns

exemplos disto seriam na verificação de regras para atendimento de créditos de certificações ambientais, integração de modelos BIM com a fabricação digital, sincronização de tabelas de prazos e custos para automatizar orçamentos 4D e 5D, criação de pontes de interoperabilidade entre software pela manutenção dos dados a serem exportados, entre muitas outras oportunidades.

Assim, o uso da programação permite uma grande flexibilidade para adequação de projetos a diferentes contextos, o que reforça a necessidade de conhecimento interdisciplinar, visto que o próprio projetista necessita conhecer tópicos específicos de sua área de interesse e como isto será programado. Além disso, a competência de programar permite agilizar e até mesmo criar novos processos de projeto relacionados às verificações automatizadas, sincronização e transferências de informações, e adequação a formatos BIM abertos, o que permite, eventualmente, atingir novos patamares de qualidade e versatilidade nos projetos, por meio do uso da computação e conhecimento interdisciplinar de forma integrada.

Assim, as consequências do desenvolvimento da competência de programar pelos próprios projetistas podem estar relacionadas ao aumento de sua independência perante as grandes plataformas, reduzindo sua necessidade de terceirizar soluções a empresas de desenvolvimento de software, e de esperar que grandes plataformas de software comercial atendam suas demandas particulares.

AGRADECIMENTOS

Agradecimentos ao PROSUP (Programa de Suporte à Pós-Graduação de Instituições de Ensino Particulares), à CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior), ao PPGARQ IMED (Programa de Pós-Graduação em Arquitetura e Urbanismo da Faculdade Meridional) pelas bolsas que possibilitaram o desenvolvimento da pesquisa; e ao Comitê Executivo Internacional da SIGraDi por prover as diretrizes de formatação.

REFERÊNCIAS

ABNT. (2013). NBR 15575-1: Edificações Habitacionais — Desempenho Parte 1: Requisitos Gerais. Rio de Janeiro, Brazil: ABNT.

- Aish, R. (2003). Extensible Computational Design Tools for Exploratory Architecture. In: Kolarevic, B. (ed), *Architecture in the digital age: design and manufacturing*. New York, Taylor & Francis Group, 23. p. 338–347.
- Aish, R., Fisher, A., Joyce, S. & Marsh, A. (2012). Progress Towards Multi-Criteria Design Optimisation Using Designscript With Smart Form, Robot Structural Analysis and Ecotect Building Performance Analysis. In: *Proceedings of the 32nd Annual Conference of the Association for Computer Aided Design in Architecture - ACADIA 12: Synthetic Digital Ecologies*. San Francisco, Estados Unidos, p. 47-56.
- Aish, R. (2013). First Build Your Tools. In: Peters B. & Peters T. (eds), *Inside Smartgeometry: Expanding the Architectural Possibilities of Computational Design*. West Sussex: John Wiley & Sons Ltd., p. 36–49.
- Aish, R. & Mendoza, E. (2016). DesignScript: a domain specific language for architectural computing. In: *Proceedings of the International Workshop on Domain-Specific Modeling - DSM 2016*. Amsterdam, Holanda p. 15–21.
- Autodesk (2017). Revit Store – Plugins, Add-ons and Extensions for Revit. <https://apps.autodesk.com/RVT/en/Home/Index>
- Celani, G., Sedrez, M., Lenz, D., & Macedo, A. (2015). The Future of the Architects's Employment: To Which Extent Can Architectural Design be Computerised? In: Celani, G., Sperling, D. M., & Franco, J. M. S. (eds), *Computer-Aided Architectural Design. The Next City - New Technologies and the Future of the Built Environment*. 16th International Conference, CAAD Futures 2015. São Paulo, Brazil, p. 195-212.
- Dresch, A., Lacerda, D. P.; & Antunes Jr., J. A. V. (2015). *Design Science Research: A Method for Science and Technology Advancement*. Springer International Publishing.
- Dynamo. (2018). DynamoBIM, Visual Programming for Revit. <http://dynamobim.org/>
- Ferreira, B. & Leitão, A. (2015) Generative Design for Building Information Modeling. In: *Proceedings of the 33rd eCAADe Conference - Volume 1*. Vienna University of Technology, Vienna, Austria, p.635-644.
- Garber, R. (2009). Optimisation Stories: The Impact of Building Information Modeling on Contemporary Design Practice. In: Garber, R. (Ed.), *Closing the Gap: Information Models in Contemporary Design Practice*. John Wiley & Sons, p. 6–13.
- Gauchat, U. (2009). The \$300,000/Year Architect. In: Garber, R. (Ed.), *Closing the Gap: Information Models in Contemporary Design Practice*. John Wiley & Sons, p. 32–37.
- Holzer, D. (2015). BIM and Parametric Design in Academia and Practice: The Changing Context of Knowledge Acquisition and Application in the Digital Age. *International Journal of Architectural Computing*, v. 13, n. 1, p. 65–82.
- Keough, I & Hauck, A. (2017). From Pencils to Partners: The Next Role of Computation in Building Design. In: Garber, R. (Ed.), *Workflows: Expanding Architecture's Territory in the Design and Delivery of Buildings*. Wiley, p. 74–81.
- Ladybug. (2018). Ladybug Tools – Home Page. <https://www.ladybug.tools/>
- Leitão, A., Santos, L., & Lopes, J. (2012). Programming Languages for Generative Design: A Comparative Study. *International Journal of Architectural Computing* 10, p. 139–162.
- Mackey, C., & Roudsari, M. S. (2018). The Tool (s) Versus The Toolkit. In *Proceedings of the Design Modeling Symposium*, Paris, France, p. 93-101.
- March, S. & Smith, G. (1995) Design and natural science research on information technology. *Decision Support Systems* 15, v. 15, n. 4, p. 251–266.
- Pittman, J. (2003) Building Information Modeling: Current Challenges and Future Directions. In: Kolarevic, B. (Ed.), *Architecture in the digital age: design and manufacturing*. New York: Taylor & Francis Group, p. 348–354.
- Prorubim. (2018) Dyno Browser Tool. <http://prorubim.com/en/tools/dyno/>
- Ottchen, C. (2009). The Future of Information Modelling and the End of Theory: Less is Limited, More is Different. In: Garber, R. (Ed.), *Closing the Gap: Information Models in Contemporary Design Practice*. John Wiley & Sons, p. 22–27.