

# AVALIAÇÃO DE DIFERENTES MÉTODOS DE AUTOMAÇÃO APLICADOS AO BIM COM BASE NA DESIGN SCIENCE

## Evaluation of Different Automation Methods Applied to BIM Based on Design Science

Victor Gonçalves dos Santos <sup>1</sup>, Maria Carolina Gomes de Oliveira Brandstetter <sup>2</sup>



### PALAVRAS CHAVE:

BIM;  
Revit®;  
Método de Automação;  
Add-in;  
Dynamo.

### KEYWORDS:

BIM;  
Revit®;  
Automation method;  
Add-in;  
Dynamo.

**RESUMO:** O processo BIM foi criado a fim de acompanhar a crescente demanda de projetos mais tecnológicos e de maior complexidade na construção civil. Em conjunto com este processo foram criados *softwares* como o Revit®, que apresentam variadas ferramentas e a possibilidade da adição de novas funções, utilizando a programação em seu código aberto. O presente artigo tem como objetivo auxiliar na escolha de uma ferramenta para a automação aplicada ao software Revit®, utilizando como metodologia a *Design Science*. É proposto um processo de quatro etapas para o desenvolvimento e avaliação de um artefato BIM. Na etapa de idealização da proposta, o artefato foi aplicado, utilizando parâmetros como o nível de conhecimento em programação e as necessidades do projeto para identificar a melhor abordagem na criação de uma rotina de automação. O método descritivo de avaliação do artefato foi utilizado, onde três cenários foram apresentados, baseando-se em situações reais acompanhadas em uma empresa do ramo de modelagem 3D e coordenação de projetos. Por fim, chegou-se à conclusão de que a proposta apresentada é válida na criação de rotinas de automação, desde que haja repetição na atividade a ser automatizada.

**ABSTRACT:** The BIM process was created in order to keep up with the growing demand for more technological and more complex projects in civil construction. In conjunction with this process, software such as Revit® was created, which features various tools and the possibility of adding new functions, using open source programming. This article aims to assist in choosing a tool for automation applied to Revit® software, using Design Science as a methodology. A four-step process is proposed for the development and evaluation of a BIM artifact. In the proposal idealization stage, the artifact was applied, using parameters such as the level of programming knowledge and the project needs to identify the best approach to creating an automation routine. The descriptive method of evaluating the artifact was used, where three scenarios were presented, based on real situations monitored in a company in the field of 3D modeling and project coordination. Finally, it was concluded that the proposal presented is valid in the creation of automation routines, as long as there is repetition in the activity to be automated.

\* Contato com os autores:

Publicado em 29 de dezembro de 2023

<sup>1</sup> e-mail: victorgs.eng@gmail.com ( V. G. dos Santos )

Engenheiro Civil pela Escola de Engenharia Civil da Universidade Federal de Goiás (EECA/UFG)

<sup>2</sup> e-mail: maria\_carolina\_brandstetter@ufg.br ( M. C. G. O. Brandstetter )

Engenheira Civil, Doutora, Docente do Curso de Engenharia Civil e Ambiental (EECA) e do curso de Pós-Graduação em Geotecnia, Estruturas e Construção Civil da Escola de Engenharia Civil da Universidade Federal de Goiás (PPG-GECON/UFG)

## 1. INTRODUÇÃO

Nas últimas décadas, a construção civil passou grandes transformações quanto aos avanços tecnológicos e diferentes formas de se analisar o processo de execução de uma obra. Com a ajuda das ferramentas CAD (*Computer-Aided Design*), projetos deixaram de ser executados manualmente e passaram a utilizar computadores para a realização dessa tarefa. Porém, com a crescente demanda de novos projetos, cada vez mais grandiosos, complexos e com novas tecnologias, novas ferramentas foram criadas para complementar a necessidade do mercado, nascendo o processo BIM (*Building Information Modeling*).

O processo possui como principais vantagens a capacidade de integrar os diversos tipos de projetos de uma edificação e a vinculação de dados aos elementos geométricos. Para manter a constante evolução do processo BIM, é essencial que os *softwares* utilizados sejam de código aberto, facilitando a troca de dados entre as partes envolvidas em um projeto, além de permitir a adição de funções especiais.

O *Revit*® é um dos *softwares* mais utilizados para a construção virtual de edificações, oferecendo uma diversa gama de ferramentas, além da possibilidade de adicionar novas funções usando ferramentas de automação. Algumas atividades no processo de desenvolvimento de um modelo 3D e da manipulação de informações dentro deste modelo podem ser bastante repetitivas e, neste contexto, podem ser inseridas novas ferramentas através da programação, utilizando a API (*Application Programming Interface*) disponibilizada pelo *Revit*®.

A *Design Science* é uma metodologia de pesquisa que possui uma filosofia prática, possuindo diretrizes que podem ser utilizadas para desenvolver e avaliar artefatos que envolvem a tecnologia BIM. Hevner *et al.* (2004) apresentam um conjunto de diretrizes que validam esta metodologia, além de facilitar o entendimento, execução e avaliação da pesquisa. A contribuição deste estudo pode ser apontada na importância do emprego da programação, capaz de oferecer ao usuário a capacidade de estender as funcionalidades dos softwares BIM existentes, encurtando processos na fase de *design* de modelos virtuais de construção.

## 2. OBJETIVO

O presente artigo tem como objetivo, utilizando como base as diretrizes da *Design Science*, apresentar e avaliar diferentes métodos de automação aplicados ao *Revit*®. Este trabalho aborda também cenários onde foi realizada a aplicação de diferentes métodos de automação, baseados em situações reais.

## 3. REVISÃO BIBLIOGRÁFICA E CONTEXTUALIZAÇÃO

### 3.1 BIM

O termo BIM tem como significado Modelagem ou Modelo de Informação da Construção (*Building Information Modeling*). Eastman *et al.* (2014) e Succar (2009) definem BIM como uma tecnologia de modelagem e um conjunto associado de processos para produzir, comunicar e analisar modelos de construção. Basicamente, espera-se que a tecnologia possa auxiliar a gerir toda a obra por meio de um modelo 3D, carregado com várias informações sobre os elementos ali modelados, como materiais utilizados, dimensões, preço, tempo de execução etc. Nesse procedimento deve-se garantir o compartilhamento e a troca de informações da obra com todas as partes interessadas.

O BIM não se reduz apenas a um *software* ou ferramenta. Trata-se de um processo, que viabiliza a tomada de decisão ao longo de todo ciclo de vida da edificação. Além da geometria da construção, abrange uma série de informações de todas as disciplinas envolvidas em um empreendimento, integrando projeto arquitetônico, estrutural, de instalações e gerencial. Um dos grandes diferenciais da utilização do BIM é que

implica em uma mudança de paradigma na modelagem para a construção civil. Isso se deve ao fato de que a representação gráfica deixa de ser abstrata e passa a possuir representação 3D onde os componentes construtivos do modelo contém semântica, ou seja, são reconhecidos pelo *software* que representam (SANTOS, 2012; SACKS *et al.*, 2018).

Biotto, Formoso e Isatto (2015) apontam como os principais benefícios da modelagem utilizando o processo BIM sendo, a facilidade na troca de informações e na comunicação e o entendimento das decisões entre os participantes, as inter-relações entre as decisões da gestão da produção e a possibilidade de simular vários cenários alternativos de forma rápida. Por outro lado, existem limitações quanto ao uso deste processo, em alguns casos tornando obrigatório o uso de outras ferramentas que permitam estender as funcionalidades dos *softwares* utilizados.

### 3.2 REVIT®

O Revit® é um *software* BIM amplamente utilizado no ramo da construção civil, criado pela Autodesk para o *design* de projetos nas mais diversas disciplinas. Este *software* tem como propósito a construção de modelos virtuais em 3D, sendo possível inserir informações paramétricas nos elementos adicionados ao modelo. Por ser um *software* BIM, existe a possibilidade do trabalho colaborativo, onde várias pessoas utilizam um modelo central, e qualquer tipo de edição realizada é automaticamente refletida em todas as vistas, como plantas, cortes e elevações.

Uma característica importante de elementos criados no Revit®, é a existência de parâmetros responsáveis por guardar as informações referentes a cada elemento e, segundo Fenato *et al.* (2018), pode-se basear nas condições de projeto para definir se os parâmetros usados serão de geometria ligados a objetos 3D, parâmetros calculados ou parâmetros de texto. Informações como estas podem alimentar tabelas e outros *softwares* ligados às várias fases de uma construção. Porém, estes valores somente poderão ser encontrados e utilizados caso estejam corretamente identificados.

#### 3.2.1 Revit® API

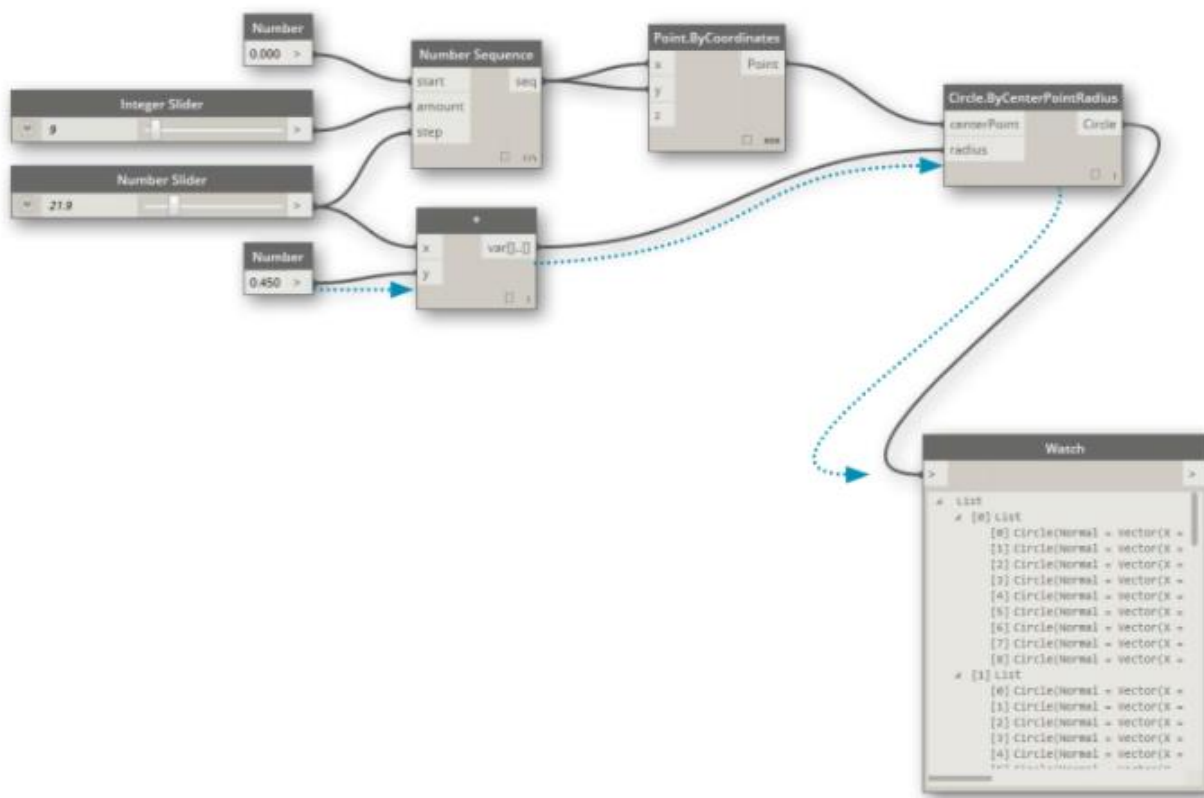
Divin (2020) define que o *software* Revit® oferece uma interface de programação de aplicativo, chamada Revit® API, que é utilizada para automatizar tarefas repetitivas, estender as funcionalidades nativas do Revit®, podendo também alterar a interface do próprio *software*. Atualmente as empresas estão sempre em busca de uma maior eficiência, e uma forma de atingir melhores resultados é a adaptação do *software* ao seu usuário, utilizando de personalizações para adequar o programa às necessidades do cliente. A automatização de tarefas repetitivas é uma das possibilidades quando se trata da extensão dos recursos do Revit®, possibilitando a junção de duas ou mais ferramentas, ou a criação de ferramentas totalmente novas. Por possuir um código aberto, pode-se usar vários tipos diferentes de programação para o desenvolvimento de rotinas de automação aplicadas ao *software*.

#### 3.2.2 Programação Visual

Dynamo é um ambiente de programação visual para BIM, ilustrado na Figura 1, que amplifica as capacidades paramétricas do Revit® utilizando de dados e lógica de um editor de algoritmo gráfico. O *software* é um projeto de código aberto para desenvolvedores e designers com o objetivo de envolver toda a comunidade de usuários na construção de ferramentas (SANDZHIEV *et al.*, 2018). No processo de programação virtual, os códigos são substituídos por nós com funções diferentes, que podem ser ligados uns aos outros das mais diversas formas, a fim de reproduzir ações baseadas na relação criada pelo usuário. Com esta ferramenta é possível fazer o processamento de dados, como extração e inserção de informações no modelo 3D, como também a criação de geometrias.

O *Dynamo* não é um programa por si só, ele é uma ferramenta inserida no *Revit*® desde 2011 para criar *scripts* dentro de um projeto, implicando que o usuário defina parâmetros e a relação entre eles (KNYAZEVA; LARIN, 2021). Da mesma forma como o *Revit*®, o *Dynamo* também possui código aberto, sendo possível adicionar novas funções a todo momento, e a comunidade de usuários está sempre tornando público novas opções de nós para serem instalados no programa.

Para programadores com conhecimento em diferentes tipos de linguagem de programação, o *Dynamo* oferece a possibilidade de criação de nós utilizando códigos, como é o exemplo dos nós de *Python*, que tornam mais abrangentes as opções oferecidas no próprio sistema. A alternativa de inserir outras linguagens de programação dentro do *Dynamo* ou outros softwares de programação visual, como o *Grasshopper*, é extremamente útil, visto que a programação visual possui limitações quanto à função dos nós, sendo necessária a inserção de códigos de programação para a execução de tarefas avançadas.



**FIGURA 1:** Programação visual apresentada pela ferramenta *Dynamo*

**FONTE:** What is Dynamo? | The Dynamo Primer (dynamobim.org)

### 3.3 DESIGN SCIENCE

A *Design Science* é uma metodologia de pesquisa utilizada para criar produtos que servem a propósitos humanos e é orientada para a tecnologia. Seus produtos são avaliados de acordo com critérios de valor ou de utilidade (MARCH; SMITH, 1995). Nos estudos de Simon (1996), é essencial para o entendimento da *Design Science*, a diferenciação das ciências natural e artificial, onde a ciência natural refere-se a um conjunto de conhecimentos voltados para os fenômenos naturais, suas características e como se comportam. Já na ciência artificial, faz-se a alusão ao que é construído pelo homem, onde artefatos são projetados com as propriedades desejadas e executam os objetivos definidos.

Kehily e Underwood (2015) ressaltam que a *Design Science* faz parte da área de pesquisa da ciência artificial, podendo ser dividida em dois tipos de pesquisa, onde o tipo 1 segue formas tradicionais de pesquisa nas universidades, onde os problemas são definidos pelos interesses intelectuais e preocupações

acadêmicas. Já o tipo 2 é mais voltado para a aplicação prática do conhecimento, onde os problemas surgem na indústria, centros de pesquisas, laboratórios e empresas.

Seguindo o tipo 2 da *Design Science*, Hevner *et al.* (2004) apresentam sete diretrizes, descritas no Quadro 1, que derivam do princípio fundamental de que conhecimento e entendimento de um problema de design e sua solução são adquiridos na construção e aplicação do artefato. Essas diretrizes foram originalmente utilizadas na avaliação do *Design Science* para sistemas de informação, porém como o BIM usa de uma plataforma totalmente digital, pode-se inserir o design de automação no mesmo contexto.

**QUADRO 1:** Diretrizes da *Design Science* adaptado para a pesquisa

Diretriz	Aplicação da Diretriz
<b>Artefato</b>	A pesquisa em <i>Design Science</i> deve produzir um artefato viável no formato de uma construção, modelo ou método.
<b>Relevância do Problema</b>	O objetivo da pesquisa em <i>Design Science</i> é desenvolver uma solução tecnológica para problemas importantes e relevantes.
<b>Avaliação do Design</b>	A utilidade, qualidade e eficácia de um artefato deve ser rigorosamente demonstrado por um método de avaliação bem executado.
<b>Contribuições da Pesquisa</b>	Pesquisas efetivas em <i>Design Science</i> devem prover contribuições claras e verificáveis nas áreas de aplicação do artefato.
<b>Rigor da Pesquisa</b>	A pesquisa em <i>Design Science</i> está embasada na aplicação de métodos rigorosos tanto na construção como na avaliação do artefato.
<b>Processo de Busca do Design</b>	A busca por um artefato efetivo requer a utilização de meios disponíveis para alcançar os fins necessários, satisfazendo os parâmetros impostos pelo problema.
<b>Comunicação da Pesquisa</b>	A pesquisa em <i>Design Science</i> deve ser apresentada efetivamente tanto para o público voltado para a tecnologia, quanto para o público voltado para o gerenciamento de projetos

**FONTE:** Adaptado de Hevner *et al.* (2004)

No artigo "*Design Science: Choosing an appropriate methodology for research in BIM*" (KEHILY; UNDERWOOD, 2015), foi analisado que diversos pesquisadores da área de *Design Science* articulam uma mesma estrutura de pesquisa, mesmo que usando diferentes terminologias. Entre os estágios em comum na metodologia geral da *Design Science*, estão o diagnóstico do problema, proposta de uma solução, implementação da solução juntamente com sua avaliação e a análise dos efeitos da aplicação.

## 4. ESTRUTURA DA PESQUISA

### 4.1 DIAGNÓSTICO DO PROBLEMA

A primeira fase da pesquisa em *Design Science* é fazer o diagnóstico do que não está funcionando ou o que precisa ser melhorado. No uso do *Revit*®, pode-se perceber que as ferramentas nativas do *software* permitem alcançar um ritmo de produção limitado, onde algumas atividades precisam ser feitas repetidamente, como a adição de informação em parâmetros ou o ajuste de elementos geométricos. Nesta etapa deve-se entender quais são as limitações impostas pelo sistema, e quais processos consomem mais tempo para serem executados.

### 4.2 PROPOSTA (DESENVOLVIMENTO DO ARTEFATO)

Hevner *et al.* (2004) propõem que a segunda etapa seja o desenvolvimento de um artefato, que irá resolver o problema prático. No caso da proposta apresentada no presente trabalho, a construção de uma solução em *Design Science* demonstra que o processo pode ser automatizado e permite a mudança em práticas de trabalho. Para que exista relevância numa perspectiva acadêmica, o processo de

desenvolvimento deve ser transparente, o que requer uma demonstração de todo o processo e das decisões tomadas que fizeram o artefato evoluir.

É nesta etapa que é realizada a criação da rotina de automação, sendo ela o artefato a ser avaliado utilizando a metodologia de pesquisa. Um importante ponto a ser analisado para o desenvolvimento de uma automação aplicada ao *Revit*<sup>®</sup> é se, para o problema diagnosticado, realmente se aplica o uso de uma rotina de automação e, caso se aplique, qual o tipo mais adequado.

Utilizando como base a análise da produção de modelos de construção virtual em um ambiente real, uma empresa do ramo, observou-se padrões na escolha da ferramenta de automação, que utilizam dois fatores para definir qual método será mais eficaz. O primeiro fator que influencia a escolha de um método de automação é o conhecimento em programação dos envolvidos na criação do artefato. Como mostrado na revisão bibliográfica, o *Revit*<sup>®</sup> possui a possibilidade de criar rotinas e adicionar funções em sua interface, podendo o usuário ter conhecimento básico ou avançado de programação. Já o segundo fator, está ligado diretamente ao diagnóstico do problema, sendo necessário entender quais são as características do problema, se é algo comum a vários projetos ou específico de um único projeto. A seguir estão listadas quatro possíveis escolhas para a criação do artefato.

#### **4.2.1 Escolha 1: Não utilizar ferramentas de automação**

Quando o problema não envolve a repetição de tarefa, ou um processo longo, a utilização de uma ferramenta de automação não se torna necessária. Além disso, deve ser analisada a influência do tempo na criação do artefato em duas situações, no caso de um usuário completamente leigo a formas de programação, exigindo que ele dedique horas de estudo para entender as ferramentas básicas da programação visual e em uma situação em que o tempo gasto para desenvolvimento do artefato seja maior que o tempo para executar a rotina manualmente utilizando as ferramentas nativas do *Revit*<sup>®</sup>.

#### **4.2.2 Escolha 2: Uso de programação visual e suas ferramentas nativas**

É recomendada a utilização do *Dynamo*, ou outro software de programação visual, para usuários que possuem conhecimento básico em programação, utilizando da programação visual para facilitar a criação do artefato. Em situações de projeto, utiliza-se o *Dynamo* para realizar funções simples, como extrair informações de parâmetros, realizar manipulação de dados e inserir informações em parâmetros. Somente utilizando as ferramentas nativas do *Dynamo* já se é possível realizar inúmeras atividades capazes de aumentar a eficiência na modelagem de elementos e no gerenciamento de dados.

#### **4.2.3 Escolha 3: Uso de programação visual e código de programação**

Em situações específicas de um projeto, onde as ferramentas nativas do *Revit*<sup>®</sup> e do *Dynamo* não são suficientes e/ou eficazes para a criação do artefato e sendo o usuário conhecedor de alguma linguagem não-visual de programação, como *Python*, *C++* e *C#*, pode-se utilizar o *Dynamo* para criar artefatos mais complexos, visto que o mesmo possui nós específicos onde podem ser inseridos código de programação utilizando dados de entrada e saída, para desenvolver rotinas que se adequem à necessidade do projeto.

#### **4.2.4 Escolha 4: Uso de código de programação para criação de *Add-in***

Neste caso é mandatório que os usuários envolvidos no processo de criação do artefato possuam conhecimento avançado em programação não visual. O uso de *Add-in* deve ser feito quando o problema diagnosticado está presente em mais de um projeto, visto que ele será inserido na interface do *Revit*<sup>®</sup>, permitindo que a execução da automação seja feita de forma fácil. Para a criação de *Add-in* s deve-se utilizar um *software* de programação externo, como o *Virtual Studio*.

### 4.3 AVALIAÇÃO E IMPLEMENTAÇÃO

Durante esta etapa, deve-se executar a implementação da rotina de programação, e traçar um mecanismo de avaliação. A utilidade, qualidade e eficiência de um artefato deve ser demonstrada por um método de avaliação bem definido, sendo através da avaliação que será medido o nível de sucesso do artefato proposto. Pode-se considerar como métodos de avaliação, entrevistas, questionários, estudos de caso, simulações através de estudos empíricos.

### 4.4 ANÁLISE DOS EFEITOS DE APLICAÇÃO

Os resultados de uma pesquisa utilizando a metodologia *Design Science* não é somente o artefato, como também o efeito que o artefato causa no ambiente em que é introduzido. Uma rotina de automação bem pensada e estruturada não somente obtém sucesso em sua execução, como também instiga a alteração de um processo existente ou a criação de novos processos, além de afetar a prática diária de trabalho. Hevner *et al.* (2004) afirmam que a pesquisa em *Design Science* deve contribuir para o conhecimento, aplicando o mesmo de uma forma inovativa.

## 5. METODOLOGIA

Uma das metodologias de avaliação do método de *design science* é a utilização de cenários, onde são criadas situações em torno do artefato para demonstrar sua utilidade. Neste trabalho todos os cenários mostram problemas e soluções reais, advindas da rotina de projetos desenvolvidas em uma empresa do ramo de modelagem de modelos virtuais e coordenação de projetos utilizando o processo BIM. Para validar a eficácia das ferramentas de automação escolhidas em cada cenário, serão aplicados os quatro estágios apresentados anteriormente.

## 6. CENÁRIOS

### 6.1 ROTINA USANDO DYNAMO PARA CALCULAR ELEVAÇÃO INVERTIDA DE TUBULAÇÕES ENTERRADAS

Para a execução da modelagem, um projeto de tubulações enterradas ao redor de um edifício escolar de 3 andares, o cliente forneceu as pranchas dos projetos hidrossanitários e, como referência para a instalação, o único valor fornecido foi a elevação invertida do tubo, identificada na Figura 2, que é a altura referente ao nível 0 de projeto da face inferior interna do tubo. Esse valor é utilizado em projetos desse tipo, pois a elevação invertida também é o nível mais baixo de uma seção de um condutor de líquido, o que determina o gradiente hidráulico disponível para mover o líquido contido no tubo.

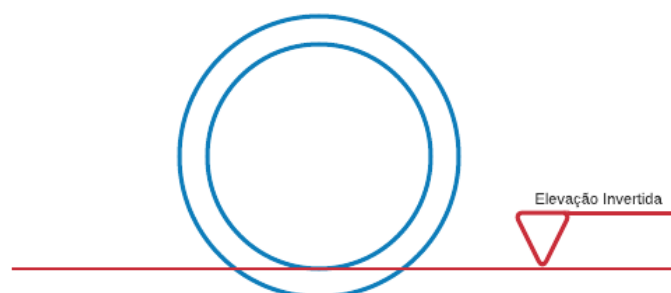


FIGURA 2: Seção axial de um tubo

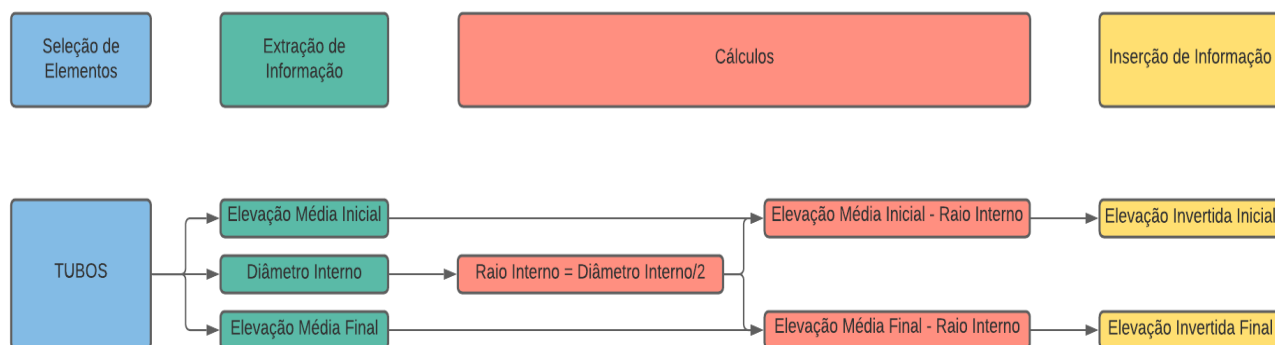
FONTE: Autoria própria.

### 6.1.1 Diagnóstico

No caso das tubulações inclinadas como, por exemplo, a drenagem de água pluvial e coleta de esgoto sanitário, o *software Revit®* informa apenas o valor médio da elevação invertida de um tubo, o que não é a melhor opção para o projeto em questão, onde é necessário conferir o valor da elevação invertida nas extremidades do tubo. Para fazer a conferência desses valores, seria necessário calcular manualmente os valores para cada tubo, o que seria inviável devido ao curto prazo para entrega do projeto.

### 6.1.2 Proposta

Por se tratar de uma situação em que, fazendo algumas simples operações matemáticas com valores já disponibilizados pelo *software*, utilizou-se da forma mais simples de automação (como citado no item 4.2.2 - Escolha 2), usando apenas a programação visual disponível no *Dynamo* para a criação de uma rotina que extraia os dados necessários, calcule o valor da elevação invertida e escreva em um parâmetro do tubo o valor para que o mesmo possa ser conferido com as pranchas. O diagrama mostrado na Figura 3 mostra o funcionamento da rotina.



**FIGURA 3:** Diagrama representativo da rotina de Dynamo para o cenário 1

**FONTE:** Autoria própria.

Inicialmente criou-se dois parâmetros compartilhados onde os valores calculados serão escritos após a rotina ser aplicada, são estes “Elevação Invertida Inicial” e “Elevação Invertida Final”. Listou-se também quais parâmetros já disponibilizados pelo sistema deverão ser utilizados para a obtenção do resultado. Foram estes, “Elevação Média Inicial”, “Elevação Média Final e Diâmetro Interno. No Quadro 2 estão listados todos os nós utilizados na construção desta rotina de automação.

**QUADRO 2:** Definição dos nós de programação visual utilizados no *Dynamo*

Nome	Definição
<b>Categories</b>	Este nó possui um menu suspenso com todas as categorias de elementos disponíveis no modelo.
<b>All Elements of Category</b>	Seleciona todos os elementos de uma categoria específica dentro do modelo.
<b>Code Block</b>	O bloco de código é universal e pode ser utilizado em inúmeras situações. Com este nó, é possível customizar funções no <i>Dynamo</i> e possibilitando também inserir códigos de programação no projeto.
<b>Element.GetParameterValueByName</b>	Este nó retorna o valor de um parâmetro dentro de um elemento desde que seja indicado o nome correto deste parâmetro.
/	Operador de divisão. Este nó retorna o resultado da divisão do valor x pelo valor y.
-	Operador de subtração. Este nó retorna o resultado da subtração do valor y do valor x.
<b>Element.SetParameterByName</b>	Este nó modifica o parâmetro de um elemento, desde que seja fornecido o nome do parâmetro e o valor a ser inserido.



FONTE: Autoria própria.

A rotina utilizada (Figura 4a) seleciona todos os elementos de uma mesma categoria (Figura 4b), neste caso tubos, extrai destes elementos os dados dos parâmetros escolhidos (Figura 4c), e para efeito de cálculo o valor do Diâmetro Interno é dividido por 2 para que seja encontrado o raio interno, e este valor é subtraído do valor dos parâmetros Elevação Média Inicial e Elevação Média Final (Figura 4d). Por fim estes valores são inseridos nos parâmetros compartilhados que foram criados antes da rotina ser desenvolvida (Figura 4e).

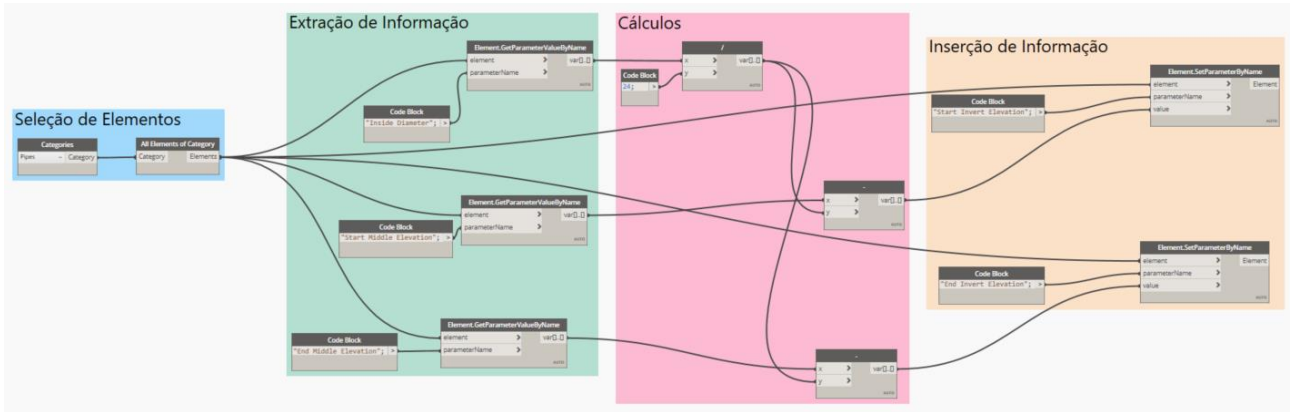


FIGURA 4a: Rotina de Automação para o Cenário 1

FONTE: Autoria própria.

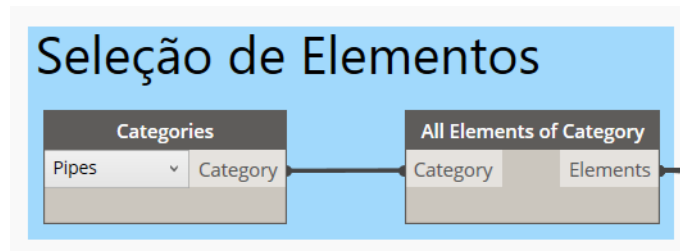


FIGURA 4b: Seleção de Elementos

FONTE: Autoria própria.

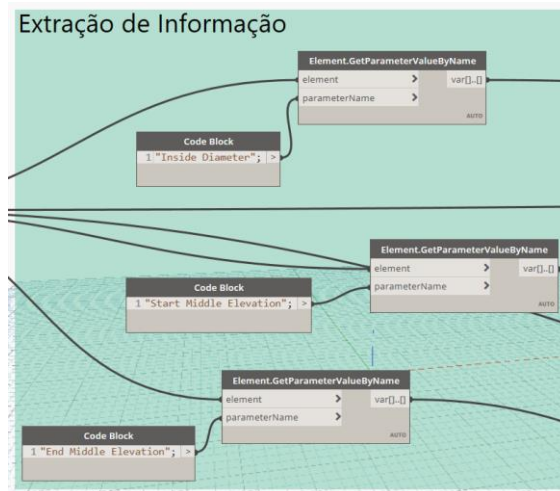
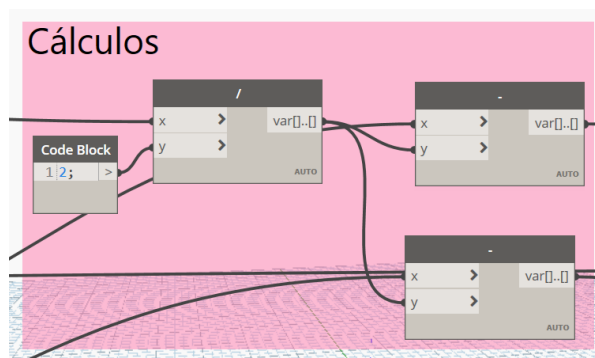
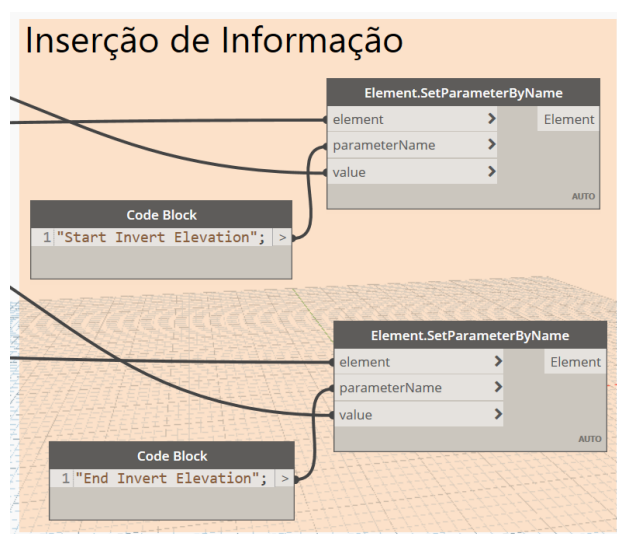


FIGURA 4c: Extração de Informação

FONTE: Autoria própria.



**FIGURA 4d:** Cálculos  
**FONTE:** Autoria própria.



**FIGURA 4e:** Cálculos  
**FONTE:** Autoria própria.

### 6.1.3 Avaliação e Implementação

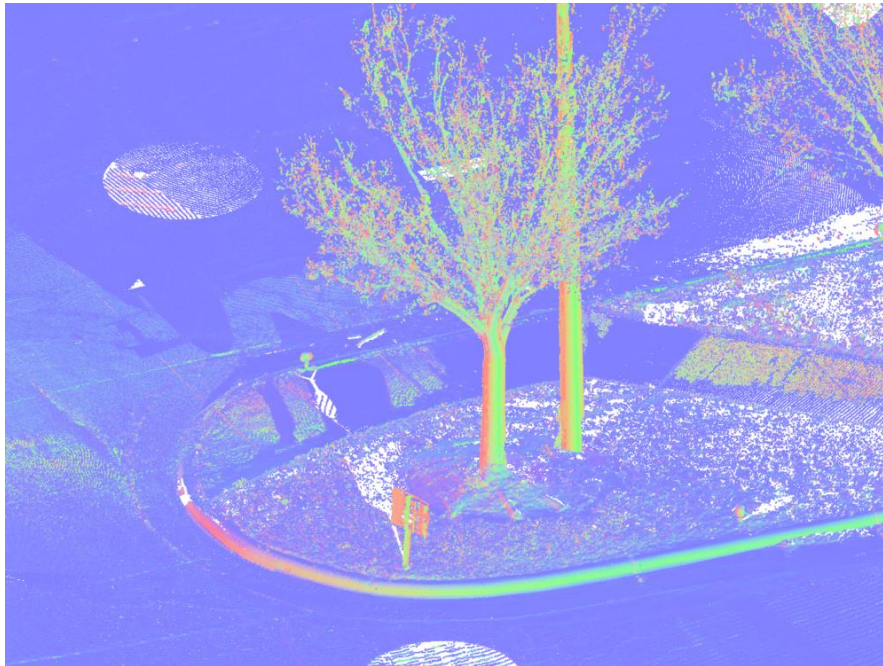
Para a avaliação desta rotina, utilizou-se de um método empírico, escolhendo tubos aleatórios no modelo com diferentes inclinações e materiais, para identificar se a rotina aplicada retorna os valores esperados. Com isso o mesmo procedimento descrito para a criação da rotina foi feito para os tubos escolhidos aleatoriamente e, após os cálculos, foi verificado que a rotina estava funcionando para todos os tipos diferentes de tubos utilizados no projeto. Com isso a rotina foi aplicada primeiramente no modo automático do *Dynamo*, mostrando não ser a forma mais adequada de se usar a rotina, visto que a cada edição em um tubo, o *software* travava por alguns segundos até a rotina ser finalizada. A forma mais eficaz para esta situação é, após a modelagem de uma quantidade maior de tubos, aplicar a rotina no modo manual, clicando na opção de executar, e automaticamente todos os tubos do projeto atualizam seus valores dos parâmetros compartilhados “Elevação Invertida Inicial” e “Elevação Invertida Final”.

### 6.1.4 Efeitos da Aplicação

O uso da rotina criada no *Dynamo* mostrou-se muito eficaz dentro do processo de modelagem do projeto, tornando um trabalho repetitivo em um procedimento automatizado, reduzindo o tempo de conferência dos valores de elevação invertida das tubulações. Após a criação desta rotina, ela foi inserida como parte do processo padrão para este tipo de projeto.

## 6.2 ROTINA USANDO DYNAMO E PYTHON PARA MODELAGEM DE MEIO-FIO

Para a modelagem *as built* da área externa de uma concessionária de veículos, foi disponibilizado para a equipe de modelagem apenas as nuvens de ponto (Figura 5), registradas por *scanner* a laser do local a ser modelado. Com estes dados, necessitou-se modelar a volumetria de toda a região escaneada, registrando elementos como postes, meio-fio, a edificação central e a elevação do terreno. A modelagem *as built* é comum para empresas que desejam ter um registro da obra finalizada ou para planejar a reforma de uma edificação. Caso a empresa ainda não tenha um modelo 3D, o uso da tecnologia de *scanners* a laser é recomendada para um levantamento preciso, gerando assim uma nuvem de pontos, que pode ser inserida no modelo de *Revit*<sup>®</sup> e servir de base para a modelagem.



**FIGURA 5:** Nuvem de pontos disponibilizada pelo cliente para a modelagem *as built*

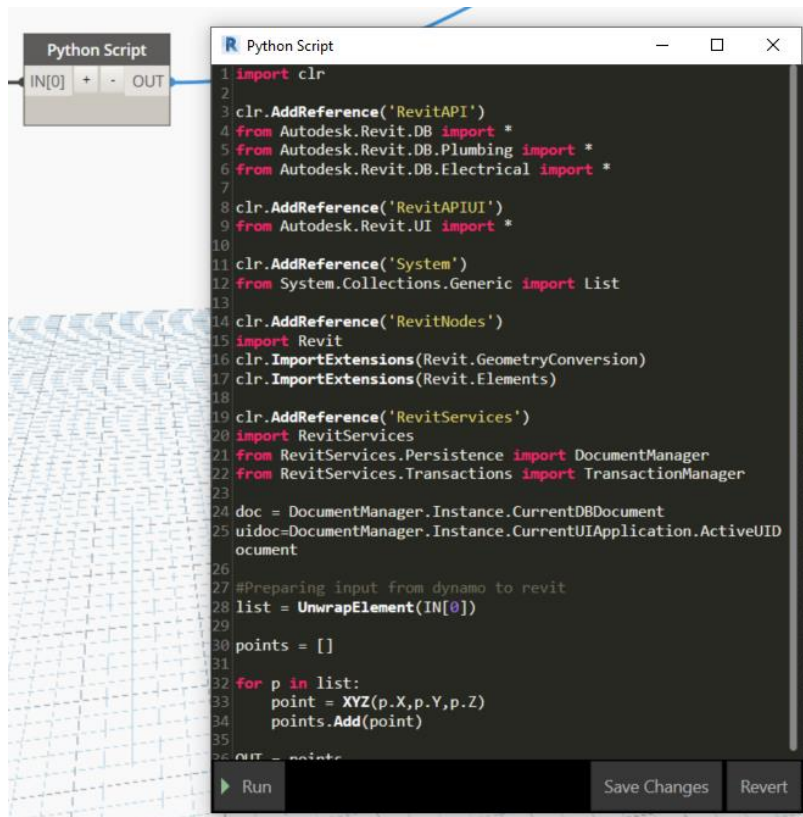
**FONTE:** Autoria própria.

### 6.2.1 Diagnóstico

Na modelagem do meio-fio, seria dispendioso construir a modelagem apenas usando as ferramentas padrão do *Revit*<sup>®</sup>, visto que elas não oferecem a possibilidade de fazer uma modelagem precisa quando se trata de elementos que possuem elevações diferentes em todo o seu comprimento. Para fazer a modelagem mais próxima da realidade deste elemento, necessitaria-se dividir o comprimento total em pequenos trechos, e modelar pequenos elementos para cada trecho.

### 6.2.2 Proposta

Nesta situação, por não haver ferramentas próprias do *Dynamo* capazes de fazer o que se pensou para a modelagem do meio-fio, utilizou-se da programação em *Python*, para criar nós específicos para esta modelagem (Figura 6) que fossem capazes de realizar a automação idealizada (como citado no item 4.2.3 - Escolha 3). Neste caso a rotina de *Dynamo* será apenas uma parte do processo de modelagem, visto que com apenas a nuvem de ponto sendo disponibilizada pelo cliente, seria necessário um tratamento prévio da nuvem e após o uso da rotina principal de *Dynamo*, uma segunda rotina seria usada para finalizar a volumetria requisitada.



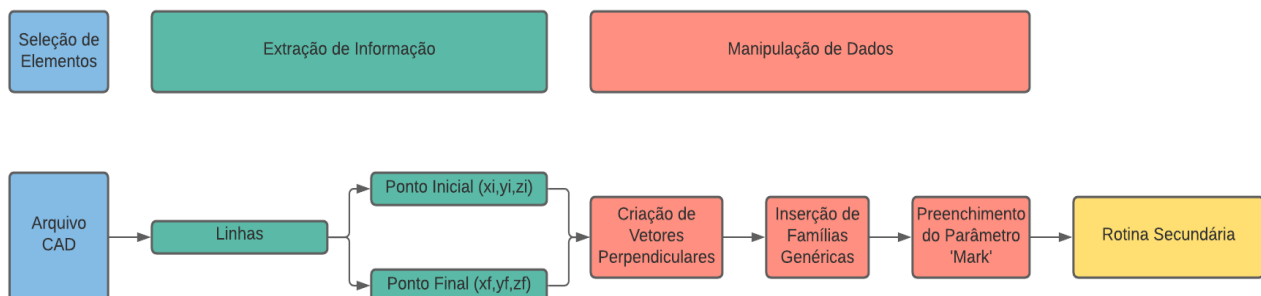
```

1 import clr
2
3 clr.AddReference('RevitAPI')
4 from Autodesk.Revit.DB import *
5 from Autodesk.Revit.DB.Plumbing import *
6 from Autodesk.Revit.DB.Electrical import *
7
8 clr.AddReference('RevitAPIUI')
9 from Autodesk.Revit.UI import *
10
11 clr.AddReference('System')
12 from System.Collections.Generic import List
13
14 clr.AddReference('RevitNodes')
15 import Revit
16 clr.ImportExtensions(Revit.GeometryConversion)
17 clr.ImportExtensions(Revit.Elements)
18
19 clr.AddReference('RevitServices')
20 import RevitServices
21 from RevitServices.Persistence import DocumentManager
22 from RevitServices.Transactions import TransactionManager
23
24 doc = DocumentManager.Instance.CurrentDBDocument
25 uidoc = DocumentManager.Instance.CurrentUIApplication.ActiveUIDocument
26
27 #Preparing input from dynamo to revit
28 list = UnwrapElement(IN[0])
29
30 points = []
31
32 for p in list:
33     point = XYZ(p.X,p.Y,p.Z)
34     points.Add(point)
35
36 OUT = points
  
```

**FIGURA 6:** Nó criado para inserir o uso de programação em Python no software *Dynamo*

**FONTE:** Autoria própria.

O objetivo da rotina principal de *Dynamo* é extrair do arquivo CAD as linhas que representam o meio-fio, criar uma lista com os pontos (x, y, z) do início e do fim de cada fração de linha, criar vetores perpendiculares e na posição destes vetores, inserir a família genérica com a seção axial do meio-fio e em cada família adicionada inserir no parâmetro “Mark” um número inteiro único, que será crucial para o funcionamento da próxima rotina de automação. Na segunda rotina, seleciona-se o arquivo de CAD e as famílias genéricas criadas, e ao aplicar a rotina, um volume será criado seguindo a ordem numérica do parâmetro “Mark”, criando assim a volumetria do meio-fio. O diagrama da Figura 7 apresenta a sequência de atividades realizada pela rotina de automação.



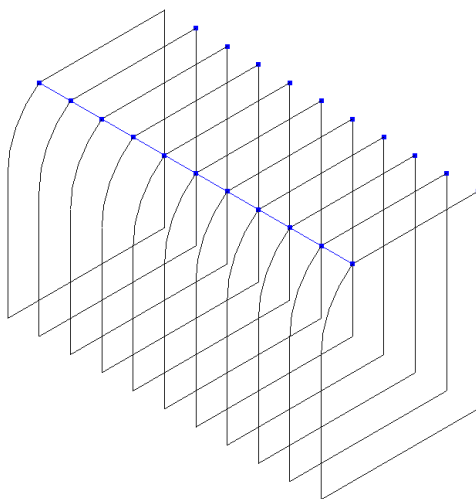
**FIGURA 7:** Diagrama representativo da rotina de *Dynamo* para o cenário 2

**FONTE:** Autoria própria.

Utilizou-se do software *Cyclone* para se fazer o tratamento da nuvem. Neste caso, isolou-se todos os pontos que representavam o meio-fio e, após isso, criou-se uma linha que representava todo o

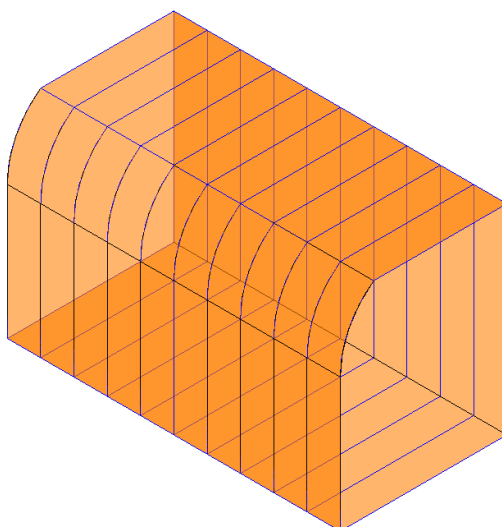
comprimento de uma das extremidades do perfil longitudinal, que será usada como referência na aplicação da rotina de *Dynamo*.

Para a execução desta rotina, é necessário que o modelador adicione um arquivo de CAD apenas com as linhas que representam o meio-fio e que seja carregado no modelo uma família genérica adaptativa representando a seção axial padrão do meio-fio. Com o arquivo CAD e a família genérica selecionadas, ao se executar a rotina principal, automaticamente as famílias genéricas foram posicionadas nos pontos de intersecção das linhas (Figura 8a). No caso de elevações diferentes, ou curvas, pode-se adicionar manualmente novas famílias genéricas ao longo da linha de referência, a fim de tornar o produto final com aparência mais suave, porém deve-se lembrar que os elementos devem estar com o parâmetro “*Mark*” com numeração em ordem crescente seguindo a ordem de posicionamento deles. Pode-se observar que com a seleção da linha de referência, das famílias genéricas e da execução da segunda rotina, instantaneamente uma volumetria (Figura 8b) é criada, que será a representação final do meio-fio no modelo 3D.



**FIGURA 8a:** Famílias genéricas inseridas no modelo

**FONTE:** Autoria própria.



**FIGURA 8b:** Volumetria gerada a partir da rotina secundária

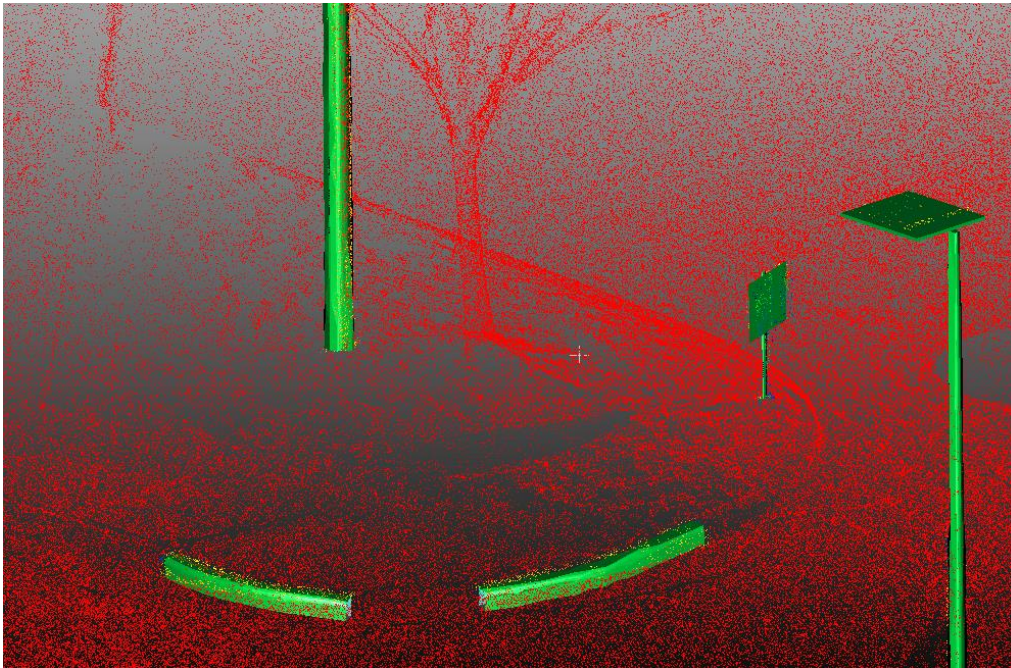
**FONTE:** Autoria própria.



### 6.2.3 Avaliação e Implementação

Como forma de avaliar a rotina descrita, utilizou-se das nuvens de pontos para se comparar com o volume gerado. Ao ativar a nuvem de pontos dentro do *Revit*®, verificou-se que o meio-fio modelado através do *Dynamo* estava visualmente adequado aos contornos mostrados nas nuvens de pontos.

Para o nível de precisão requisitado para o projeto, apenas a avaliação visual foi suficiente para validar o funcionamento da rotina, porém existem ferramentas como os *softwares Cloud Compare* e *Cyclone* que utilizam de comparação entre os elementos no *Revit*® e as nuvens de pontos, gerando uma nova nuvem de pontos, onde cada ponto é colorido baseado na proximidade dos elementos do *Revit*®.



**FIGURA 9:** Comparação entre nuvem de ponto e modelo 3D realizada no *Cloud Compare*

**FONTE:** Autoria própria.

### 6.2.4 Efeitos da Aplicação

Para o projeto aplicado, a rotina foi bastante efetiva, acelerando a modelagem dos elementos requisitados, sem reduzir a qualidade do modelo. O maior efeito da aplicação da modelagem usando uma rotina de *Dynamo* com programação em *Python*, não foi somente para este projeto, visto que proporcionou a empresa a possibilidade de inovar quando se trata da resolução de problemas no processo de modelagem. Este método aumenta o número de possibilidades no uso do *software Revit*®, já que com a programação em *Python*, pode-se estender as funcionalidades do mesmo, expandindo a usabilidade do *software* na modelagem e no gerenciamento da informação.

## 6.3 ADD-IN EMPREGANDO A LINGUAGEM DE PROGRAMAÇÃO C# PARA EXPORTAÇÃO DE PONTOS DE CONTROLE

Na instalação de suportes para tubulações, dutos de ar-condicionado e equipamentos em edifícios multi pavimentos, utiliza-se em campo, um equipamento capaz de local o exato ponto de inserção do suporte, baseando-se em coordenadas reais retiradas do modelo 3D. A exportação destes pontos de inserção costuma ser feita por um *software* pago, que possui limitações ao se tratar do gerenciamento de informações.

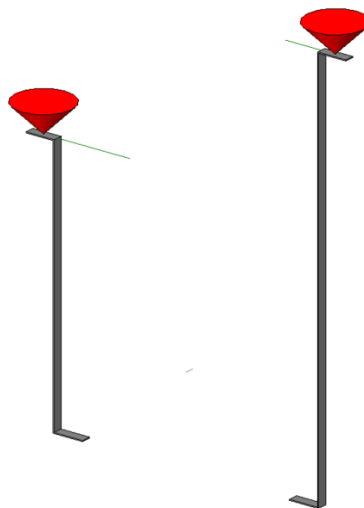
### 6.3.1 Diagnóstico

Neste caso, não existe necessariamente um problema com o modelo 3D e as ferramentas do *Revit*<sup>®</sup>, porém a ferramenta atualmente disponível no mercado requer um alto investimento financeiro, então pensou-se em criar uma versão paralela deste *software* usando alguma forma de programação. O objetivo é localizar os pontos de inserção de todos os suportes do modelo, e extrair o nome de cada um dos pontos, juntamente com as coordenadas x, y e z, em um formato de arquivo que seja lido pelo equipamento utilizado em obra para realizar a locação dos pontos.

### 6.3.2 Proposta

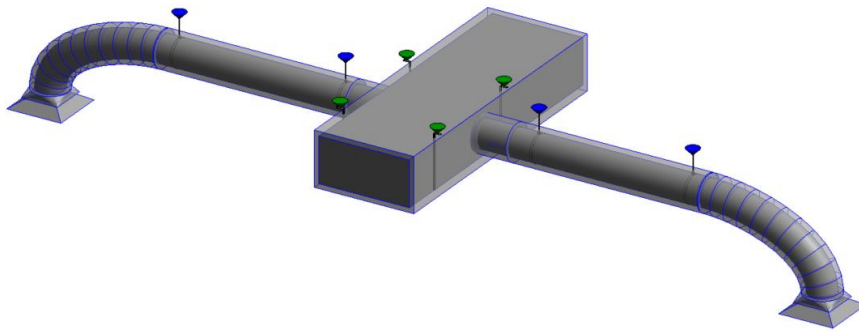
Por ser um processo realizado em diferentes projetos, optou-se por usar a linguagem de programação C# e o *software Virtual Studio* para a criação de um *Add-in* que faça parte da interface do *Revit*<sup>®</sup> em qualquer versão e possa ser usado em qualquer projeto de maneira simples (como citado no item 4.2.4 - Escolha 4). Para que o *Add-in* funcione corretamente, é necessário adicionar uma família como componente aninhado, que sirva como ponto de controle dentro das famílias de suporte utilizadas no projeto. Após o posicionamento e coordenação de todos os suportes dentro do modelo 3D, utilizaria-se o *Add-in* para que as informações de todos os pontos sejam encontradas no modelo, permitindo assim a exportação dos pontos em um arquivo em formato csv (*Comma-separated values*).

Inicialmente, utilizou-se as ferramentas nativas de edição de famílias para adicionar cones nos pontos de inserção dos suportes nas lajes de concreto (Figura 10a). A extremidade deste cone representa um ponto com coordenadas x, y e z que serão extraídas com o uso do *Add-in* desenvolvido. Com as famílias de suporte devidamente configuradas, pode-se inserir todos os suportes necessários no modelo (Figura 10b), podendo utilizar-se de uma rotina simples de *Dynamo* para dar uma nomenclatura específica para cada um dos suportes e seus pontos de inserção.



**FIGURA 10a:** Instalação de cone como componente aninhado na família de suporte

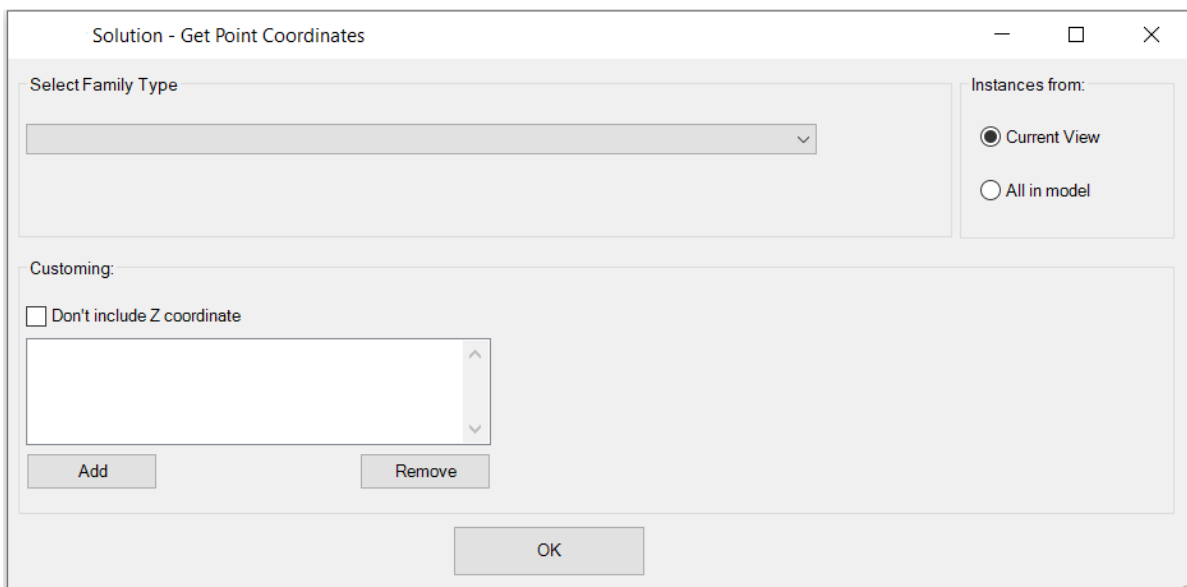
**FONTE:** Autoria própria.



**FIGURA 10b:** Suportes inseridos no projeto de dutos de ar-condicionado

**FONTE:** Autoria própria.

Após realizada a coordenação de todas as disciplinas de projeto, garantindo as posições finais dos suportes e com o *Add-in* devidamente instalado, na aba de *Add-ins*, em “Ferramentas Externas”, escolhe-se o nome do *Add-in* criado e automaticamente uma nova janela aparecerá (Figura 10c). Dentro desta janela, seleciona-se o nome da família do cone aninhado nas famílias de suporte, após é feita a decisão se serão exportadas as informações de todos os elementos do modelo ou somente os selecionados na vista atual, por fim adiciona-se os elementos na janela para serem exportados e aperta-se OK para concluir a atividade.



**FIGURA 10c:** Interface do *Add-in* criado

**FONTE:** Autoria própria.

O produto final é gerado instantaneamente na forma de um arquivo csv (Figura 10d) contendo o nome de cada ponto e suas coordenadas para instalação dos suportes na obra. Para garantir que o processo ocorrerá sem erros, deve-se garantir que não existam 2 elementos com o mesmo nome, que o nome dos elementos não seja muito longo e que exista um processo para controlar a numeração dos pontos, garantindo que cada elemento tenha um código único e que não se repita ao longo da modelagem mesmo que o elemento seja futuramente removido.



	A	B	C	D
1	HANGER_ID	X COORDINATE	Y COORDINATE	Z COORDINATE
2	SQ-L01-A2-1	-74.64025408	-10.70779402	10.17486877
3	SQ-L01-A2-2	-74.64025408	-7.124460692	10.17486877
4	SQ-L01-A2-3	-66.64025408	-10.70779402	10.17486877
5	SQ-L01-A2-4	-66.64025408	-7.124460692	10.17486877
6	SQ-L01-A2-5	-58.64025408	-10.70779402	10.17486877
7	SQ-L01-A2-6	-58.64025408	-7.124460692	10.17486877
8	SQ-L01-A2-7	-50.64025408	-10.70779402	10.17486877
9	SQ-L01-A2-8	-50.64025408	-7.124460692	10.17486877
10	SQ-L01-A2-9	-46.64025408	-10.70779402	10.17486877
11	SQ-L01-A2-10	-46.64025408	-7.124460692	10.17486877
12	SQ-L01-A2-11	-38.64025408	-10.70779402	10.17486877
13	SQ-L01-A2-12	-38.64025408	-7.124460692	10.17486877
14	SQ-L01-A2-13	-30.64025408	-10.70779402	10.17486877
15	SQ-L01-A2-14	-30.64025408	-7.124460692	10.17486877
16	SQ-L01-A2-15	-25.89025408	-10.70779402	10.17486877
17	SQ-L01-A2-16	-25.89025408	-7.124460692	10.17486877

**FIGURA 10d:** Informações extraídas pelo *Add-in*

**FONTE:** Autoria própria.

### 6.3.3 Avaliação e Implementação

Como forma de avaliar o *Add-in*, foi realizada uma comparação com o *software pago*, utilizando dois modelos idênticos e aplicando o processo com o *Add-in* criado na empresa e com o *software* externo. Os dois apresentaram a mesma informação com precisão equivalente e mesmo nível de qualidade. Com isso foi comprovada a eficácia do produto, podendo assim ser implementado.

Para a implementação do *Add-in* precisou-se repensar o processo, visto que o *software pago* possui ferramentas adicionais que não foram incorporadas ao *Add-in*. Uma delas é o controle da nomenclatura dos pontos, para que não haja pontos com o mesmo nome, e para que esta nomenclatura siga algum padrão. A forma alternativa parece este controle, foi o uso de uma rotina simples de *Dynamo*, capaz de preencher um parâmetro compartilhado com um código alfanumérico único para cada um dos pontos de controles.

### 6.3.4 Efeitos da Aplicação

Além de exportar os dados dos pontos de controles de forma instantânea, a ferramenta criada na própria empresa não oferece nenhum custo adicional, ao contrário do *software* terceirizado, trazendo assim um benefício econômico, além de apresentar a possibilidade de utilização do *Add-in* em vários projetos, o que pode ser feito com outros processos comuns entre projetos.

## 7. CONCLUSÕES

O artigo apresenta uma proposta de escolha para o método de automação aplicado ao Revit®, baseando-se nas diretrizes da *Design Science* e, utilizando o método descritivo, apresenta três cenários que validam o objetivo proposto. O artefato criado é produto de um bom diagnóstico do problema, a análise de fatores que influenciam na escolha do método de automação, a implementação e avaliação do artefato desenvolvido e pelo esclarecimento do impacto causado pela aplicação do artefato em um contexto de prática de trabalho.

Observa-se com os cenários, que todas as diretrizes da *Design Science* foram seguidas e que efetivamente existe a criação de um artefato, onde os problemas diagnosticados possuem uma relevância no contexto do projeto, os métodos de avaliação são capazes de demonstrar a utilidade, qualidade e eficácia do artefato, além de verificada a contribuição clara do artefato no contexto do projeto, onde os meios disponíveis são capazes de alcançar os fins necessários, respeitando as limitações impostas.

Por fim, conclui-se que a proposta apresentada é válida na criação de rotinas de automação, desde que haja repetição na atividade a ser automatizada, e que o usuário dedique tempo para adquirir, no mínimo, conhecimentos básicos em programação visual. Espera-se que este artigo seja capaz de auxiliar o público interessado em otimizar o processo de produção de modelos de construção virtual, como também instigar pesquisadores a propor novos processos, criar novos artefatos e gerar conhecimento para o aprimoramento deste campo de pesquisa.

## 8. REFERÊNCIAS

- BIOTTO, C. N.; FORMOSO, C.T.; ISATTO, E. L. **Uso de modelagem 4D e Building Information Modeling na gestão de sistemas de produção em empreendimentos de construção**. Ambiente Construído, Porto Alegre, v. 15, n. 2, p.65-77, 2015.
- DIVIN, N. V. BIM by using Revit API and Dynamo. A review. **Alfabuild**, v. 14, Article nº 1404, 2020.
- EASTMAN, C.; TEICHOLZ, P.; SACKS, R.; LISTON, K. **Manual de BIM: Um Guia de Modelagem da Informação da Construção para Arquitetos, Engenheiros, Gerentes, Construtores e Incorporadores**. São Paulo: Bookman, 2014.
- FENATO, T. M.; SAFFARO, F. A.; BARISON, M. B; HEINECK, L. F. M.; SCHEER, S. Método para elaboração de orçamento operacional utilizando um *software* de autoria BIM. **Ambiente Construído**, Porto Alegre, v. 18, n. 4, p. 279-299, 2018.
- HEVNER, A. R.; MARCH, S. T.; PARK, J.; RAM, S. Design Science in information system research. **MIS Quarterly**, v. 28, n. 1, p. 75-105, 2004.
- KEHILY, D.; UNDERWOOD, J. Design Science: Choosing an appropriate methodology for research in BIM. In: CITA BIM Gathering, 2015, Dublin. **Proceedings...** Dublin: 2015.
- KNYAZEVA, N.; LARIN, V. Automation of repetitive tasks when designing an eco-friendly residential home. **E3S Web of Conferences**, v. 244, 2021.
- SACKS, R.; EASTMAN, C.; LEE, G.; TEICHOLZ, P. **BIM Handbook: a Guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors**. Third ed. New Jersey: Wiley, 2018.
- SANDZHIEV, N. V.; LALIN, V. V.; SAVCHENKO, A. V.; SERDIUKOV, D. A. Dynamo platform for automation Revit. **Alfabuild**, v. 6, n. 7, p. 75-82, 2018.
- SUCCAR, B. Building Information Modeling Framework: A research and delivery foundation for industry stakeholders. **Automation in Construction**, v. 18, n. 3, p. 357-375, 2009.