



UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Civil Arquitetura e Urbanismo

CAUE CARNEIRO SANTIAGO

**DESAFIOS TECNOLÓGICOS DA VERIFICAÇÃO
AUTOMÁTICA DE MODELOS BIM**

CAMPINAS

2023

CAUE CARNEIRO SANTIAGO

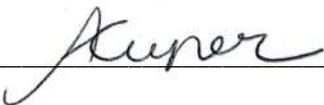
**DESAFIOS TECNOLÓGICOS DA VERIFICAÇÃO
AUTOMÁTICA DE MODELOS BIM**

Dissertação de Mestrado apresentada a Faculdade de Engenharia Civil, Arquitetura e Urbanismo da Unicamp, para obtenção do título de Mestre em Arquitetura, Tecnologia e Cidade, na área de Arquitetura, Tecnologia e Cidade.

Orientadora: Profa. Dra. Ana Regina Mizrahy Cuperschmid

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA DISSERTAÇÃO DEFENDIDA PELO ALUNO CAUE CARNEIRO SANTIAGO E ORIENTADO PELA PROFA. DRA. ANA REGINA MIZRAHY CUPERSCHMID.

ASSINATURA DA ORIENTADORA



CAMPINAS

2023

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Rose Meire da Silva - CRB 8/5974

Sa59d Santiago, Caue Carneiro, 1990-
Desafios tecnológicos da verificação automática de modelos BIM / Caue Carneiro Santiago. – Campinas, SP : [s.n.], 2023.

Orientador: Ana Regina Mizrahy Cuperschmid.
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Civil, Arquitetura e Urbanismo.

1. Modelagem de informação da construção. 2. Projeto arquitetônico. 3. Planejamento urbano. 4. Plano diretor. I. Cuperschmid, Ana Regina Mizrahy, 1975-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Civil, Arquitetura e Urbanismo. III. Título.

Informações Complementares

Título em outro idioma: Technological challenges of BIM-based automated code compliance checking

Palavras-chave em inglês:

Building information modeling

Architectural design

Urban planning

Master plan

Área de concentração: Arquitetura, Tecnologia e Cidade

Titulação: Mestre em Arquitetura, Tecnologia e Cidade

Banca examinadora:

Ana Regina Mizrahy Cuperschmid [Orientador]

Norberto Corrêa da Silva Moura

Max Lira Veras Xavier de Andrade

Data de defesa: 12-04-2023

Programa de Pós-Graduação: Arquitetura, Tecnologia e Cidade

Identificação e informações acadêmicas do(a) aluno(a)

- ORCID do autor: <https://orcid.org/0009-0005-3851-5557>

- Currículo Lattes do autor: <http://lattes.cnpq.br/0607243267625594>

UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Civil Arquitetura e Urbanismo

**DESAFIOS TECNOLÓGICOS DA VERIFICAÇÃO AUTOMÁTICA DE
MODELOS BIM**
CAUE CARNEIRO SANTIAGO

Dissertação de Mestrado aprovada pela Banca Examinadora, constituída
por:

.....
Profa. Dra. Ana Regina Mizrahy Cuperschmid
Presidente e Orientadora / UNICAMP

.....
Prof. Dr. Norberto Corrêa da Silva Moura
USP

.....
Prof. Dr. Max Lira Veras Xavier de Andrade
UFPE

A Ata da defesa com as respectivas assinaturas dos membros encontra-se
no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa
da Unidade.

Campinas, 12 de abril de 2023

RESUMO

Sistemáticas BIM para otimização de processos, vem sendo amplamente desenvolvidas nos últimos anos, tendo em vista os benefícios que elas proporcionam para o setor de Arquitetura, Engenharia e Construção (AEC). Um dos usos BIM que merecem destaque é o *code checking* ou verificação de normas, que permite a automatização de verificações específicas e pertinentes à diversas modalidades de projetos, como checagem de parâmetros urbanísticos, atrelados aos Códigos de Obras ou às Leis de Uso e Ocupação do Solo. Tendo isso em vista, torna-se imperativo a busca por meios de se automatizar os processos de verificação de normas. Portanto, busca-se o desenvolvimento de um sistema que permita a verificação de forma automática de modelos BIM, buscando uma melhor compreensão dos desafios tecnológicos envolvidos, usando como base a Lei de Uso e Ocupação de Solo da cidade de Poços de Caldas. Essa pesquisa se desenvolve por meio de uma *Design Science Research* (DSR). O principal artefato desenvolvido é uma Instanciação, a qual permite a criação de sistema de validação de modelos IFC arquitetônicos, com base na Lei de Uso e Ocupação da cidade de Poços de Caldas, Minas Gerais. Por meio de uma Revisão Sistemática da Literatura (RSL) foi possível identificar softwares e os processos utilizados para checagem automática de regras. A partir disso, foi desenvolvida uma rotina via linguagem visual de programação (VPL) que realiza a checagem de modelos IFC. Esse sistema permite a validação de parâmetros urbanísticos como recuo frontal, afastamentos laterais e de fundo, altura máxima da edificação, coeficientes de permeabilidade, ocupação e aproveitamento, de forma automática, apresentando ao usuário um relatório final. Durante o desenvolvimento do artefato, foram observadas diversas limitações e desafios tecnológicos. A sua explicitação gera contribuições para novas iniciativas de elaboração desse tipo de ferramenta. No futuro, espera-se que ajustes na ferramenta sejam feitos para aumentar a sua abrangência.

Palavras-chave: Verificação de normas; Verificação de Modelos; BIM; IFC; Projeto Arquitetônico

ABSTRACT

Title: Technological Challenges of BIM-Based Automated Code Compliance Checking

BIM systematics for process optimization has been widely developed in recent years, due to the benefits they provide for the Architecture, Engineering and Construction (AEC) sector. One of the BIM uses worth mentioning is code compliance checking or rule checking, which allows the automation of specific checks relevant to different types of projects, such as checking urban parameters, linked to the Works Codes or the Land Use and Occupation Laws. With this in mind, it becomes imperative to search for ways to automate the project verification processes. Therefore, the aim is to develop a system that allows the automatic verification of BIM models, seeking a better understanding of the technological challenges involved, based on the Land Use and Occupation Law of the city of Poços de Caldas. This research is developed through a Design Science Research (DSR). The main artifact developed is an Instantiation, which allows the creation of a validation system for architectural IFC models, based on the Law of Use and Occupation of the city of Poços de Caldas, Minas Gerais. Through a Systematic Literature Review (SLR) it was possible to identify software and processes used for automatic checking of rules. From this, a routine was developed via visual programming language (VPL) that performs the checking of IFC models. This system allows validation of urban parameters such as frontal setback, lateral and background spacing, maximum height of the building, permeability coefficients, occupation and use, automatically, presenting the user with a final report. During the development of the artifact, several limitations and technological challenges were observed. Its explanation generates contributions to new initiatives for the elaboration of this type of tool. In the future, adjustments to the tool are expected to be made to increase its coverage.

Keywords: Code Compliance Checking; Model Verification; BIM; IFC; Architecture Design

LISTA DE FIGURAS

Figura 1	Modelo de apresentação de projeto.....	19
Figura 2	Etapas para checagem automática de regras.....	21
Figura 3	IFC e MVD.....	24
Figura 4	Diferentes abordagens.....	26
Figura 5	Diferenças entre linguagens.....	29
Figura 6	Níveis de uma checagem.....	30
Figura 7	<i>Strings</i> de busca.....	36
Figura 8	Artigos Aprovados.....	38
Figura 9	Ano de Publicação.....	39
Figura 10	Publicações por Periódico.....	39
Figura 11	Coocorrência das palavras-chave.....	40
Figura 12	Propósito do artigo.....	43
Figura 13	Implementação e Avaliação de <i>Software</i>	44
Figura 14	Discussão de Dados Semânticos.....	46
Figura 15	Propósito da Checagem.....	47
Figura 16	Abordagem.....	48
Figura 17	Softwares de Modelagem.....	50
Figura 18	Uso do IFC.....	51
Figura 19	Software de checagem.....	52
Figura 20	Dificuldades encontradas.....	53
Figura 21	Camadas do processo de desenvolvimento do artefato.....	57
Figura 22	Fluxograma das etapas da pesquisa.....	60
Figura 23	Implantação.....	64
Figura 24	Estudo 1 – Criação de Polígonos.....	65
Figura 25	Estudo 1 – Rotação e Transladação de Geometria.....	66
Figura 26	Estudo 1 – União dos sólidos.....	67
Figura 27	Estudo 1 – Verificação de Intersecção.....	67
Figura 28	Estudo 2 – Geometria base.....	68
Figura 29	Estudo 2 - Diagramação.....	69
Figura 30	Estudo 3 – Sólidos base criados.....	70

Figura 31	Estudo 3 – Diferença entre os sólidos.....	70
Figura 32	Estudo 3 – Geometria importada.....	71
Figura 33	Modelo IFC importado	72
Figura 34	Tentativa de importação de IFC	73
Figura 35	Nós de seleção e entrada de dados – Data-Shapes.....	73
Figura 36	Inserções de informações para melhoria de IU.....	74
Figura 37	Janelas de (a) seleção de vínculo e (b) seleção de elementos do vínculo.....	75
Figura 38	Substituição de parte do código fonte do nó	75
Figura 39	Malha criada.....	76
Figura 40	Poligonais (a) antes do ajuste e (b) após ajuste	77
Figura 41	Envelopamento dos afastamentos	78
Figura 42	Sintaxe da área trazida para Dynamo	79
Figura 43	(a) Filtro para seleção de modelo genérico e (b) modelo trazido ao Dynamo.....	80
Figura 44	Pré-teste – fluxo do processo de validação de modelo BIM.....	81
Figura 45	Visão geral do artefato no ambiente Dynamo.....	83
Figura 46	Grupo Zoneamento.....	85
Figura 47	Grupo Área do terreno e <i>offset</i> do plano.....	87
Figura 48	Grupo de malha e linhas perimetrais de afastamentos.....	89
Figura 49	Grupo Área Construída, mesma sistemática do pré-teste.....	90
Figura 50	Grupo Área Construída	91
Figura 51	Grupo de Área Permeável.....	92
Figura 52	Grupo Recuo Frontal	94
Figura 53	Sintaxe da verificação numérica	96
Figura 54	Condiciona da verificação numérica e nó <i>Watch</i> com “É saída” selecionado.....	97
Figura 55	Sintaxe da verificação geométrica.....	98
Figura 56	Condiciona da verificação geométrica.....	99
Figura 57	Interface do Usuário, resultados.....	100
Figura 58	Implantação-fluxo do processo de validação de modelo BIM.....	101

Figura 59 Demonstração de possível falha de reconhecimento de recuo frontal.....105

SUMÁRIO

1 INTRODUÇÃO.....	12
1.1 OBJETIVO	15
1.1.1 Objetivo geral.....	15
1.1.2 Objetivos específicos	15
1.2 CONTEXTUALIZAÇÃO	16
1.2.1 Iniciativas para aprovação de projetos em BIM em prefeituras.....	17
1.2.2 Processo de aprovação de projetos em Poços de Caldas.....	18
1.3 ESTRUTURA DA DISSERTAÇÃO	19
2 FUNDAMENTAÇÃO TEÓRICA	21
2.1 CHECAGEM AUTOMÁTICA DE REGRAS EM BIM.....	21
2.1.1 Etapas para checagem da automática de regras	21
2.1.2 Abordagens para checagem automática de regras.....	26
2.1.3 Níveis de checagem.....	30
2.1.4 Softwares utilizados para checagem automática de regras.....	31
2.2 REVISÃO SISTEMÁTICA DA LITERATURA	35
2.2.1 Busca, elegibilidade e codificação	37
2.2.2 Avaliação da qualidade	42
2.2.3 Síntese dos resultados.....	46
2.2.4 Considerações.....	54
3 MATERIAIS E MÉTODOS	57
3.1 MÉTODO.....	57
3.2 MATERIAIS	60
4 PROJETO DO ARTEFATO	62
4.1 PROPOSIÇÃO DO ARTEFATO.....	62
4.2 OBJETO DO ESTUDO.....	63
4.3 VERIFICAÇÃO DE MODELO BIM: ESTUDOS	64
5 DESENVOLVIMENTO DO ARTEFATO	72
5.1 PRÉ-TESTE	72

5.2 IMPLEMENTAÇÃO.....	82
5.2.1 Zoneamento	83
5.2.2 Área do terreno e offset do plano.....	86
5.2.3 Criação de malha e linhas perimetrais e de afastamento	88
5.2.4 Área construída.....	90
5.2.5 Área permeável e Projeção	92
5.2.6 Recuo frontal.....	93
5.2.7 Resultados.....	95
5.3 DIRETRIZES DE UTILIZAÇÃO	101
6 AVALIAÇÕES	104
6.1 AVALIAÇÃO DO ARTEFATO.....	104
6.2 AVALIAÇÃO DOS DESAFIOS TECNOLÓGICOS	105
7 DISCUSSÃO E CONCLUSÃO	108

1 INTRODUÇÃO

A adoção do *Building Information Modeling* (BIM) cresce constantemente no cenário mundial, haja visto seu potencial no setor da construção (HOCHSCHEID; HALIN, 2019). Nos últimos cinco anos, passa por grandes avanços e conta com fluxo constante de crescimento de empresas inovadoras, que possuem como especialidade a aplicação de diversas tecnologias de automação e informação (SACKS; GIROLAMI; BRILAKIS, 2020a).

Fica claro que o governo brasileiro entende a importância do BIM, traçando estratégias para sua disseminação a nível federal. O decreto presidencial nº 9.983 de 2019 (Brasil, 2019), institui o Comitê Gestor da Estratégia do *Building Information Modelling* e também dispõe sobre a Estratégia BIM-BR. Corroborando, o decreto nº 10.306 de 2020 (Brasil, 2020) estabelece três fases para sua disseminação. As fases 1 e 2 trazem que o BIM deverá ser utilizado para obras de grande relevância na execução direta ou indireta de obras e serviços de engenharia. Já na fase 3, o mesmo deverá ser aplicado para obras de média ou grande relevância.

Soma-se ao fato a Lei de Licitações e Contratos Administrativos, LEI nº 14.133, de 1º de abril de 2021, onde indica em seu capítulo II, Da Fase Preparatória, Seção I, Da Instrução do Processo Licitatório, Art. 19, inciso V, parágrafo 3º que deverá ser preferencialmente adotado o BIM para licitações de obras e serviços de engenharia e arquitetura (BRASIL, 2021).

O estado de Santa Catarina, foi um dos pioneiros no desenvolvimento de estímulos ao BIM, tendo desenvolvido um Caderno de Especificações de Projetos em BIM, ilustrando os procedimentos que devem ser adotados nas licitações para contratação de projetos e “*As Built*” pelo Estado no âmbito do Poder Executivo (SANTA CATARINA, 2020). O caderno faz diversas especificações com relação ao desenvolvimento dos modelos e às boas práticas de projeto, como a definição de uma coordenada espacial comum entre as diversas disciplinas.

Com isso, tem-se observado um aumento na implementação BIM no desenvolvimento de projetos no Brasil. Entretanto, por mais que tenha sido adotado como prática na elaboração de projetos ou licitações, poucas são as ações notadas no sentido de automatizar o processo de verificação de projetos. No Brasil, o uso do BIM para checagem automática de regras com foco na aprovação de projetos de prefeitura, mostra-se incipiente.

Em relação às esferas públicas municipais, existem desafios específicos para sua implementação. É necessário que existam sistemas ou interfaces que facilitem a aprovação de projetos tão como a capacitação dos funcionários, adequação da infraestrutura, do parque tecnológico, estímulo à adesão e treinamento dos profissionais.

Para viabilizar o processo de aprovação digital de projetos, diversas ferramentas vêm sendo desenvolvidas e utilizadas (DYNAMO BIM, 2021). Dentre os usos do BIM, encontra-se o *code checking* ou verificação de normas, que pode ser aplicada nas mais diversas disciplinas e para várias finalidades, como por exemplo, na checagem de normativas de segurança contra incêndio e pânico (KINCELOVA et al., 2020). O *code checking* visa verificar as conformidades do modelo em relação às regras impostas. A checagem de modelos, utilizando sistemáticas de automação, aumentam consideravelmente a eficácia dos processos de projeto, bem como reduzem o tempo com verificações manuais (TREBBI et al., 2020).

A checagem de regras pode ser aplicada num aspecto muito mais amplo, desde normativas de órgãos públicos, bem como instruções específicas de clientes e organizações privadas. De forma geral, o escopo da automação de regras se enquadra nas categorias (SOLIHIN; EASTMAN, 2015b):

- Verificação da boa formação de um modelo;
- Checagem de normativas da construção;
- Exequibilidade e outros requerimentos do contratante;
- Segurança e outras regras que possibilitem ações corretivas;
- Aprovação de garantias (seguros);

- Integralidade dos dados BIM para entrega à gerência das instalações.

No entanto, a checagem automática de regras precede o próprio BIM (SOLIHIN; EASTMAN, 2015c). Segundo os autores, o assunto tem sido alvo de pesquisas desde a década de 60. Atualmente, diversas pesquisas buscam entender os mecanismos e processos para desenvolvimento de sistemas que façam a checagem automática de regras em BIM. Vários deles lidam com a semântica envolvida nos processos, fazendo uso de sistemáticas como PAS e NLP, entre outros (ZHANG; EL-GOHARY, 2017; SONG et al., 2020). Outros trabalhos lidam com a criação de ferramentas específicas, apresentando um caso de uso, também utilizando diversas abordagens (FAROOQ; SHARMA; KUMAR, 2018; KANG, 2020). Percebe-se que mesmo com diversos trabalhos realizados nos últimos anos e com as diversas metodologias abordadas, ainda não existe uma solução ideal que seja abrangente o suficiente para as diversas especificidades do setor da Arquitetura, Engenharia e Construção (AEC).

Essa pesquisa parte do pressuposto que parâmetros urbanísticos podem ser incorporados em ferramentas de checagem automática de regras para aprovação digital de projetos em BIM. A partir disso, as questões norteadoras são:

- Que sistemas podem ser identificados no processo de checagem automática de regras em BIM?
- Quais as abordagens voltadas à checagem automática de regras são as mais utilizadas?
- Quais as principais dificuldades encontradas nesse processo?
- Para que os projetistas possam realizar a verificação automática de regras em BIM, o uso do formato IFC seria um pré-requisito?
- É possível desenvolver um sistema customizável para aprovação digital de projetos em BIM, sem um amplo conhecimento em linguagem de programação de software?

- Como o sistema poderia ser incorporado na aprovação digital de projetos em BIM?

O desenvolvimento de sistemas automatizados BIM, que verificam parâmetros urbanísticos para a construção de edificações, especialmente na fase de estudo preliminar, proporcionaria agilidade, eficiência e confiabilidade ao setor. Além disso, garantiria a avaliação de projetos de acordo com as leis de uso e ocupação do solo.

1.1 OBJETIVO

1.1.1 Objetivo geral

Busca-se uma melhor compreensão dos desafios tecnológicos inerentes da verificação automática de normas, imersas em um contexto BIM. Para tanto, é desenvolvido um sistema para permitir que os projetistas verifiquem automaticamente modelos BIM, que estejam em fase de Estudo Preliminar, tendo em vista a Lei de Uso e Ocupação do Solo (LUOS). Para contextualizar a pesquisa e permitir experimentações, é utilizada a legislação da Prefeitura do Município de Poços de Caldas, MG.

1.1.2 Objetivos específicos

- Identificar as principais ferramentas, processos e dificuldades encontradas na criação de mecanismos de verificação automatizada de regras;
- Desenvolver um artefato, a título de experimentação, no qual o projetista seja capaz de verificar modelo BIM de acordo com um determinado conjunto de regras da LUOS;
- Avaliar se o artefato desenvolvido é capaz de trazer resultados confiáveis, atendendo as normativas selecionadas;

- Discutir os desafios da criação de um sistema de checagem automatizada de regras, visando a aprovação de projetos de prefeitura.

1.2 CONTEXTUALIZAÇÃO

Tradicionalmente, a submissão de projetos para aprovação em prefeitura exige diversas documentações – IPTU, Escritura, ART ou RRT, além do próprio projeto em si. Toda a documentação passa pelos trâmites de entrada do projeto junto ao órgão ou secretaria específicos, muitas vezes, esse processo é realizado de forma presencial. No cenário nacional, foram instaurados em algumas localidades, processos para submissão dos arquivos de forma *online*, trazendo certa praticidade tanto ao proprietário, como ao responsável técnico.

Na Prefeitura Municipal de São Paulo, instaurou-se um sistema eletrônico de aprovação de projetos (SÃO PAULO, 2022a). Trata-se dos mesmos trâmites usualmente feitos em papel, porém de forma eletrônica, sendo que os arquivos de projeto são entregues em formato .DWF e os demais documentos em formato .PDF ou .JPG. O sistema de *upload* de informações, apesar de *online*, tem funcionamento limitado entre segundas e sextas-feiras, das 7:00 às 22:00 (SÃO PAULO, 2022b).

Já na Prefeitura Municipal do Rio de Janeiro, o cadastro inicial para submissão do projeto é feito de forma *online*, mas os demais trâmites são feitos de forma impressa, devendo ser entregues presencialmente (RIO DE JANEIRO, 2022). De forma semelhante, a Prefeitura Municipal de Campinas faz exigências de diversos documentos impressos, como requerimento padrão, plantas do projeto simplificado, ficha informativa cadastral, ART ou RRT, entre outros (CAMPINAS, 2022). Toda a documentação impressa deve ser levada pelo responsável técnico ou proprietário até o setor específico da prefeitura, para que sejam feitos os trâmites de aprovação.

Observa-se que a maioria das prefeituras no Brasil ainda não aderiram à digitalização do processo de submissão de projetos e

permanecem com os mesmos processos manuais, com impressão da documentação necessária e submissão dos arquivos de forma presencial, em setores específicos da prefeitura.

1.2.1 Iniciativas para aprovação de projetos em BIM em prefeituras

No contexto global de aprovação digital de projetos, se tem como referência Singapura, com sua plataforma de submissões automatizadas, a *CORENET e-Submission System*. A plataforma possui um sistema baseado em internet entre o governo e o setor privado, em que é possível realizar a submissão de arquivos para as autoridades reguladoras para aprovação de projetos (SINGAPORE GOVERNMENT, 2016). O esquema IFC é utilizado como base para as representações dos envios, uma vez que por ser neutro, pode ser extraído de diversas fontes distintas, permitindo a elaboração mais livre e independente por parte dos projetistas (BCA, 2022).

A aplicação de práticas que possam otimizar o fluxo de aprovações de projetos de prefeitura vem sendo vislumbradas em alguns centros do Brasil, como em Salvador, com o projeto Salvador 360 (SEDUR, 2018). O Projeto inclui diversos eixos de atuação, sendo que o seu carro chefe é o Salvador 360 Simplifica (JAIMES, 2017). Tal eixo do programa tem a premissa de simplificar a obtenção de licenciamento de empresas, publicidade e obras. O eixo responsável pela verificação de modelos IFC, denomina-se Metropolis (METROPOLIS SOLUTION, 2020). Seu objetivo principal é otimizar a gestão de licenciamento de construções, desburocratizando e agilizando o processo de análise de conformidade (TORCH ENGENHARIA, 2022).

De acordo com site oficial, o sistema faz uma estratificação dos dados do esquema IFC, inseridos no modelo autoral. Tal modelo, de forma inicial, deve ser elaborado utilizando-se o software Revit, para que as checagens sejam realizadas. Ressalta-se que toda a documentação necessária para aprovação de projetos, como escrituras, ART ou RRT, matrícula, plantas dos projetos detalhadas, entre outros, ainda devem ser submetidas, de forma digital, na plataforma

1.2.2 Processo de aprovação de projetos em Poços de Caldas

O processo de aprovação de projetos de construção na cidade de Poços de Caldas, segue como base as prescrições legais diretas de decretos e leis complementares. A Lei Complementar nº 92, dispõe sobre o uso e ocupação do solo, norteia os cidadãos no que se refere aos zoneamentos da cidade, bem como os grupos de uso, recuos frontais, afastamentos laterais de fundo, e demais parâmetros urbanísticos, como taxa de ocupação, coeficiente de aproveitamento e taxa de permeabilidade (Poços de Caldas, 2007), Anexo A.

Para a coesão das informações necessárias, o Decreto nº 11.687 dispõe sobre a padronização e apresentação de projetos em seu Capítulo I, Da Apresentação de Projeto, Art. 3º, Figura 1, (Poços de Caldas, 2015). O Decreto ainda traz em seu Art. 2º do mesmo capítulo, todos os documentos complementares para o início dos trâmites de aprovação, Anexo B.

Todos os documentos devem ser entregues de forma impressa, no ato da abertura do protocolo, sendo feito de forma presencial. Os trâmites internos seguem o que indica o Art. 7º do Capítulo II, Dos Procedimentos Internos, Anexo C. Dentro de um prazo de seis meses, da data de aprovação do projeto, se não for emitido o Alvará de Construção, o projeto perde sua validade, sendo necessário o início de todo o trâmite novamente.

O capítulo 4, PROJETO DO ARTEFATO, demonstra qual será a proposta do artefato, o objeto de estudo e apresenta estudos iniciais para o que foi proposto.

O capítulo 5, DESENVOLVIMENTO DO ARTEFATO, faz o pré-teste e implementa o artefato dentro de um contexto real, demonstrando suas especificidades e limitações.

O capítulo 6 AVALIAÇÕES, traz à tona a avaliação do artefato, mostrando o que foi desenvolvido, as limitações e conflitando com os resultados esperados.

Por fim, o capítulo 7, DISCUSSÃO E CONCLUSÃO, relata as conclusões obtidas no contexto do desenvolvimento do artefato.

2 FUNDAMENTAÇÃO TEÓRICA

Para iniciar o embasamento e busca pelo estado da arte sobre aprovação digital de projetos, é feita uma fundamentação teórica. Serão descritas as características dos processos de checagem automática de regras em BIM, o uso do esquema IFC dentro do universo de checagem e quais os softwares comerciais mais utilizados.

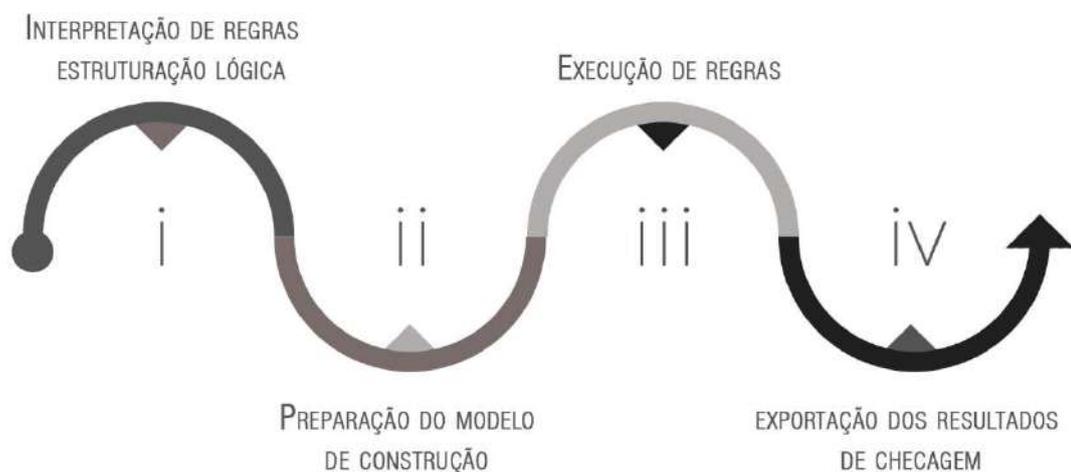
2.1 CHECAGEM AUTOMÁTICA DE REGRAS EM BIM

Serão tratados a seguir as etapas da checagem de regras, as possíveis abordagens e níveis.

2.1.1 Etapas para checagem da automática de regras

No contexto de checagem automática de regras, é possível identificar quatro etapas para sua implementação, (i) interpretação de regras e estruturação lógica; (ii) preparação do modelo de construção; (iii) execução de regras; e (iv) exportação dos resultados de checagem (EASTMAN et al., 2009), Figura 2.

Figura 2 Etapas para checagem automática de regras



Fonte: Autor

a) Interpretação de regras e estruturação lógica

As regras, leis, decretos e outros documentos legais que trazem orientações e restrições, com relação à aprovação de projetos, sejam eles em quaisquer âmbitos, são geralmente textos criados e representados em linguagem natural, isto é, não possuem necessariamente um léxico e estruturação lógica passíveis de serem interpretados por máquinas sem que modificações sejam feitas (SOLIMAN-JUNIOR; FORMOSO; TZORTZOPOULOS, 2020).

Um dos grandes desafios encontrados nessa etapa encontra-se justamente na transcrição de regras escritas em linguagem natural, para uma linguagem passível de ser reconhecida e interpretada pela máquina, também chamado de tradução de linguagem natural (LN) para linguagem de máquina (LM) (LEE; CHIANG, 2016). As regras escritas e desenvolvidas de forma natural, trazem muitas vezes, informações não explícitas. Assim, precisam ser traduzidas para uma linguagem que possa ser interpretada pelos sistemas computadorizados (FAROOQ; SHARMA; KUMAR, 2018; HOSSAIN; AHMED, 2019; NAWARI, 2019; SCHWABE; TEIZER; KÖNIG, 2019).

Uma linguagem intermediária que faz o mapeamento de regras naturais e as transpõe para a linguagem de máquinas é a lógica *first-order-predicate* ou predicado de primeira ordem (EASTMAN et al., 2009). O predicado é uma expressão que atribui uma propriedade ao objeto identificado pela variável da sentença, sendo que a lógica demonstrada trata de realizar substituições devidas, em trechos de sentenças, para a validação do predicado, como verdadeiro ou falso, lidando também com sentenças de quantificação (O'LEARY, 2016).

Frases sem valores afirmativos que faltam as devidas atribuições como “*é um número inteiro*” não possuem a devida atribuição e por sua vez, não podem ser validadas. Só com a inclusão de um parâmetro inicial é que podem ser feitas afirmações do tipo verdadeiro-falso para a sentença. Se for incluído o número “4”, então a sentença é verdadeira, pois “*4 é um número inteiro*” corresponde à realidade. Do mesmo modo, se for incluído o número

“3,14”, então a sentença é falsa, pois “3,14 é um número inteiro” não corresponde à realidade.

Outra forma de se criar uma estruturação lógica é através da ontologia de nomes e propriedades (EASTMAN et al., 2009). O termo ontologia foi emprestado da filosofia, que trata da natureza da existência e para sistemas automatizados ou de inteligência artificial, tudo que é pode ser representado (GRUBER, 1995).

A sistemática pode ser definida como uma conceitualização de um domínio específico (ex.: engenharia civil ou engenharia de software), que provê afirmações lógicas e explícitas sobre classes, instâncias e propriedades, e traduz as regras em linguagem natural para linguagem de máquinas (ZHONG et al., 2018). De forma simplificada, há uma interface entre o sistema de ontologia, que possui uma biblioteca de informações inicial e as instâncias do modelo (ZHONG et al., 2012). A partir disso, é realizado a união entre os pares de informações, para que então a frase em linguagem natural consiga ser transposta com clareza para linguagem de máquina e ser passível de utilização pelos sistemas de checagem. Um exemplo seria a ontologia “*OWL: cardinality (n)*” que busca informações sobre o número de valores que determinada propriedade possui.

b) Preparação do modelo de construção

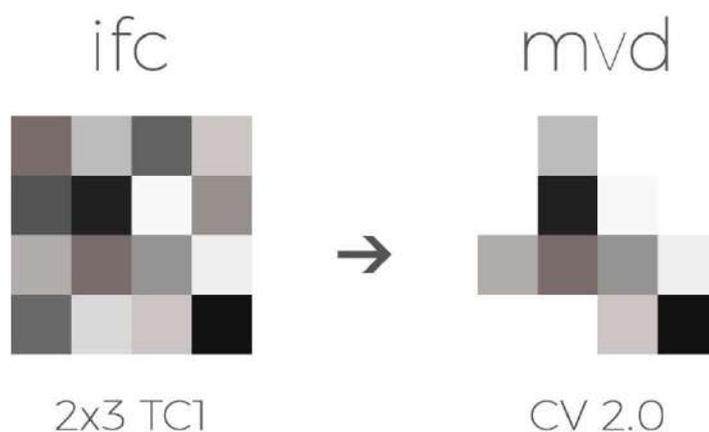
As necessidades atuais, no que diz respeito aos requerimentos de um modelo, são muito superiores às necessidades relativas à um desenho bidimensional (EASTMAN et al., 2009). Segundo os autores, antes era necessário que a representação tivesse semelhança visual com o que se pretendia passar. No entanto, dentro do universo BIM, cada objeto possui embutido em si, informações específicas e pertinentes ao seu tipo ou classe e não somente a sua aparência visual.

O *Industry Foundation Class (IFC)* é o esquema neutro dentro do universo BIM (BUILDING SMART INTERNATIONAL, 2022a). Seu objetivo é a possibilidade da definição de especificações das informações ou dados de troca entre vários setores de conhecimento (LEE, 2015). O esquema IFC

permite a troca de dados entre os diversos setores da AEC, trazendo à tona a possibilidade de interoperação (BUILDING SMART INTERNATIONAL, 2022b). Para estabelecer uma comunicação eficiente entre os diversos componentes do setor, uma vez que cada um possui as suas especificidades, se tem a necessidade da criação de um subconjunto de informações, que compõe o arquivo IFC (LEE; SOLIHIN; EASTMAN, 2019). Esse subcomponente é denominado como *Model View Definition* (MVD).

Um MVD é um subconjunto do esquema IFC que tem como intuito facilitar o fluxo de trabalho BIM, por selecionar certas especificações mais representativas para cada uso, Figura 3 (BUILDING SMART INTERNATIONAL, 2022c).

Figura 3 IFC e MVD



Fonte: Autor

A criação de MVD's sem a devida padronização, pode levar a incompatibilidade entre os modelos IFC, gerando assim problemas de visualização e falta de informações (BUILDING SMART INTERNATIONAL, 2022c). Para contornar esse problema, mecanismos orientados aos objetos são desenvolvidos, através da criação de objetos de modelo ou construção melhorados. Essa abordagem funciona para a grande maioria dos casos, mas pode não ser suficiente para conjuntos mais complexos de configurações espaciais (EASTMAN et al., 2009).

A criação de projetos, que derivam de modelos-base, ou *templates* também é uma alternativa para tentar facilitar a preparação de seus requisitos, com relação à checagem automática de regras (EASTMAN et al., 2009). Um *template* pode ser desenvolvido e customizado, baseado nos critérios específicos para cada caso, podendo conter parâmetros ideais para os procedimentos de checagem (SEGHIER et al., 2017).

c) Execução de regras

A fase de execução de regras faz o papel de unir as etapas anteriores, do sistema de regras estruturado e do modelo devidamente consolidado (EASTMAN et al., 2009). Ainda segundo os autores, tendo as regras sido transcritas de linguagem natural para linguagem de máquinas e as conexões entre tais informações estejam coerentes com as informações contidas no modelo, tem-se então um processo simples e objetivo, em que são feitas as devidas checagens e o sistema pode ser testado.

O modelo pode também passar por uma checagem anterior, para verificação de erros, omissões, interferências e definições contraditórias, para posteriormente passar pelo crivo da checagem automática de regras (BAZJANAC, 2005). Essa pré-checagem pode ser feita por softwares autorais, comerciais externos ou sistemas autorais criados especificamente para a situação.

Desde 2009, tem-se apontado que, por muitos anos, as revisões de código seriam uma mistura de verificação automática e manual, em função da implementação e lançamento incremental das regras base, advindas de leis, decretos, etc (EASTMAN et al., 2009). Atualmente, observa-se que os usos do BIM passaram por avanços significativos (LI; CAI; KAMAT, 2016; SACKS; GIROLAMI; BRILAKIS, 2020b; SCHRANZ; URBAN; GERGER, 2021). Contudo, no que se refere à aprovação de projetos BIM de forma automatizada, é perceptível um certo atraso com relação aos desenvolvimentos de plataformas, sistemas, etc.

d) Exportação dos resultados de checagem

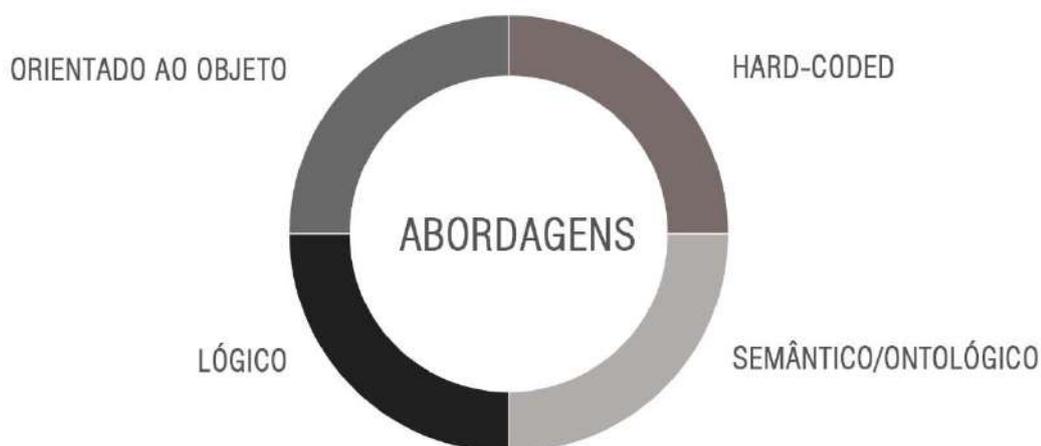
A última etapa do processo de criação de um sistema de checagem automática de regras é o relatório de checagem. As condições impostas pelas regras – em primeira etapa – caso satisfeitas, devem receber um retorno positivo, do contrário devem receber um retorno negativo (BURGGRÄF et al., 2021). A forma como esse relatório será mostrado ao usuário pode sofrer alterações, podendo aparecer de formas gráficas ou textuais, mostrando quais regras não foram cumpridas.

Fica claro que a etapa de interpretação de regras e estruturação lógica, especialmente no que se refere à tradução de regras de linguagem natural para linguagem de máquinas é uma etapa crucial para o desenvolvimento de uma sistemática funcional de checagem de regras, sendo essa a principal barreira para uma implementação mais substancial desse uso BIM (SONG et al., 2020).

2.1.2 Abordagens para checagem automática de regras

Foi possível observar quatro tipos de abordagem para implementação de sistema de checagem de regras, sendo elas: (a) *hard-coded*; (b) orientado ao objeto; (c) lógico; e (d) semântica / ontológica, Figura 4 (SOLIMAN-JUNIOR; FORMOSO; TZORTZOPOULOS, 2020).

Figura 4 Diferentes abordagens



Fonte: Autor

a) *Hard-coded*

A abordagem *hard-coded* faz uso de softwares ou sistemas de checagem comerciais e tem a conveniência de ser mais prática, uma vez que dentro de tais softwares já vem embutidas diversas regras pré-programadas (GUO; ONSTEIN; LA ROSA, 2021). Os softwares comerciais mais conhecidos para essa abordagem são: Solibri Model Checker (SMC), SMARTreview, FORNAX (CORENET System), Jotne Express Data Manager (EDM) Model Checker, entre outros.

Entretanto, devido ao grande esforço manual na extração de informações pertinentes de textos legais, os sistemas *hard-coded* são menos escaláveis (JIANG et al., 2019) e consequentemente não alcançam um processo totalmente automatizado (ZHANG; EL-GOHARY, 2017). O fato de serem programas que: (i) usam abordagens *black-box*, isto é, não deixam claro ao usuário como o processo de checagem foi realizado; (ii) possuem um alto custo de aquisição; (iii) são inflexíveis com relação à modificações e; (iv) carecem de estruturas generalistas, que possam ser aplicadas em diversas situações e diversas modalidades do setor da AEC, contribuem para a utilização de outras abordagens de checagem (NAWARI, 2019).

Para alguns autores, abordagens *black-boxes* não devem ser levadas adiante, num contexto de pesquisas, uma vez que não é possível a verificação das análises e a reflexão crítica dos processos torna-se falha (BURGGRÄF et al., 2021).

b) Orientada ao objeto

A abordagem orientada ao objeto, também chamada de baseada em objetos, se direciona à objetos de construção e objetos de regras (GUO; ONSTEIN; LA ROSA, 2021). Uma vantagem dessa abordagem em relação à abordagem *hard-coded*, está na possibilidade da criação de regras de tal modo que a seleção dos elementos pode ser alternada conforme necessidade do usuário, gerando assim uma maleabilidade maior do sistema (SCHWABE; TEIZER; KÖNIG, 2019).

Por haver a possibilidade da criação de regras e o controle sobre os processos de checagem, as vantagens sobre mecanismos *black-boxes* são grandes, uma vez que ela torna-se mais apropriada à customização (BURGGRÄF et al., 2021). De modo geral, a abordagem orientada aos objetos, quando aliada com linguagens de programação visual (ex.: Dynamo, Grasshopper, Marionette) pode facilitar o entendimento da lógica da programação da checagem (PREIDEL et al., 2016; KIM et al., 2019). Soma-se ao exposto, que ela vem sendo gradualmente adotada em conjunto com *machine learning*, a qual vem sendo alvo de pesquisas na área (GUO; ONSTEIN; LA ROSA, 2021).

c) Lógica

Já a abordagem lógica, se baseia em duas partes distintas, sendo elas, a conversão de LN para LM e a obtenção de um modelo BIM devidamente detalhado (NAWARI, 2013). Dentro do espectro de checagem de regras, sistemas inteligentes como Inteligência Artificial (IA) podem ter um papel importante, no que se refere ao enriquecimento semântico de partes do código conceito, definidos no MVD, enriquecimento de relações topológicas, até a checagem de regras, utilizando o *Deep Learning* (SACKS et al., 2019).

Para a tradução de regras de ordem natural, algumas pesquisas apontam para o método *Requirement, Applicability, Select e Exception* ou RASE. A metodologia trabalha com marcadores para palavras específicas de um texto, atribuindo um dos quatro operadores possíveis, criando então uma estrutura passível de ser interpretada por máquinas (HJELSETH; NISBET, 2011).

Outra possibilidade para a abordagem lógica, está na utilização do *Building Environment Rule and Analysis* ou BERA, que trata-se de uma linguagem de programação de domínio específico, Figura 5, que tem o intuito de analisar e acessar regras que dizem respeito ao ambiente construído (LEE, 2011).

Figura 5 Diferenças entre linguagens

	LINGUAGEM DE MODELAGEM	LINGUAGEM DE PROGRAMAÇÃO
LINGUAGENS DE USO GERAL	XML UML IDEF ORM	C JAVA C++ VB C# ... PYTHON
LINGUAGENS DE DOMÍNIO ESPECÍFICO	GBXML CIS/2 RVT/DGN ... IFC	SSQL SCL BERA

Fonte: Adaptado de (LEE, 2011)

Essas e outras formas de lidar com a tradução de regras de LN para LM, focam na estrutura das regras propriamente dita e não lidam com os problemas de interpretação semântica que as regras naturais trazem consigo (SOLIHIN, 2016).

d) Semântica / ontológica

Conceitos básicos de modelos de construção e suas relações, podem ser descritos através de uma modelagem ontológica. A semântica *web* e tecnologias ontológicas já são utilizadas no setor de AEC (GUO; ONSTEIN; LA ROSA, 2021). Uma vez que o IFC não descreve todas as relações e interdependências entre os elementos de construção, a ideia do seu enriquecimento semântico torna-se atrativa (SACKS et al., 2017).

Tal abordagem pode facilitar a troca de informações, integrações e otimizar a habilidade de colaboração entre sistemas distintos (NACER; AISSANI, 2014). Para o setor de AEC promover o uso de abordagens semânticas *web*, a BuildingSMART padronizou uma ontologia ifcOWL. A partir daí, protocolos como SPARQL e de linguagens de consulta ou *query*

languages, como RDF ou *Semantic Web Rule Language* (SWRL) para extração de dados começa a se tornar mais difundido (GUO; ONSTEIN; LA ROSA, 2020).

2.1.3 Níveis de checagem

Existem quatro classes ou níveis paramétricos voltadas para a checagem automática de regras, e são eles: (1) regras que requerem uma pequena quantidade ou apenas um dado específico; (2) regras que requerem valores simplesmente derivados (ou de simples derivação); (3) regras que necessitam de uma extensa estrutura de dados; e (4) regras que necessitam de “prova de solução” ou que criam a solução, Figura 6 (SOLIHIN; EASTMAN, 2015a).

Figura 6 Níveis de uma checagem



Fonte: Autor

No nível 1, as regras checam atributos explícitos de componentes do modelo e são usadas em geral para verificar se determinados elementos possuem os atributos corretos embutidos em si, simples checagem como tipologia dos elementos, se atributos semânticos estão corretos, como nomes de objetos, dentre outros (SOLIHIN; EASTMAN, 2015a).

Ainda segundo os autores, no nível 2 são realizadas as verificações que requerem valores simples ou um pequeno conjunto de valores

derivados, mas que não criam nenhuma estrutura de dados nova. São feitos cálculos trigonométricos, no qual as relações entre objetos distintos são buscadas, como por exemplo, a distância mínima entre portas ou se determinado ponto encontra-se dentro de uma poligonal. É um nível em que a indústria terá grande benefício com avanços de aplicações.

Em seguida, no nível 3, as regras necessitam de uma estrutura de dados extensa, englobando dados de construção mais complexos, com níveis semânticos maiores. Os programas desenvolvidos para verificação de regras nesse nível, dependem de análises tanto geométricas como topológicas. Geralmente, para lidar com correlações complexas entre objetos, é necessário que a aplicação possua uma biblioteca de modelos (sólidos).

No quarto e último nível, as regras não somente devem possuir como saída um relatório de observância (*output*), isto é, se passaram ou não em determinada verificação, mas sim, requerem uma “prova de solução”. Essa classe, geralmente, está mais em busca de uma solução para determinado problema do que da verificação em si, sendo mais utilizada quando existe risco de fatalidade, aplicada diretamente à segurança, especialmente em canteiros de obra.

2.1.4 Softwares utilizados para checagem automática de regras

Atualmente, existem softwares comerciais capazes de implementar aspectos referentes à checagem de regras. As variações entre eles encontram-se em sua flexibilidade de adaptação aos processos de checagem, na modelagem de informações de projetos, na codificação de códigos de construção ou regras específicas, entre outros.

As ferramentas comerciais Solibri Model Checker, SMARTreview, FORNAX e Jotne Express Data Manager Model Checker são apontadas como as mais utilizadas para checagem automática de regras (GUO; ONSTEIN; LA ROSA, 2021). Adicionalmente, observa-se o uso de softwares de linguagem visual de programação como Dynamo, Grasshopper e Marionette

(GHANNAD et al., 2019). Esses softwares são brevemente apresentados a seguir.

- Solibri Model Checker (SMC)

O SMC é um programa BIM que verifica modelos de arquitetura e engenharia, no que se refere à qualidade, integridade e segurança (GRAPHISOFT, 2022). Possui arquitetura base em JAVA, no qual é feita a leitura de arquivos IFC e uma série de funções embutidas, como bibliotecas de pré-chechagem de modelo, verificações de formas sobrepostas, convenções de atributos, existência de objetos, saída de emergência, distância de rota de fuga, entre outros (NAWARI, 2013).

De acordo com o site da empresa, o programa pode ser adquirido via assinatura, com valores de 280 Euros por mês e de 1.870 Euros por ano. Para outras opções, é necessário entrar em contato com a empresa (SOLIBRI E-COMMERCE, 2022). O software possui diversos benefícios, por já ter embutido em si várias possibilidades de verificação. Contudo, por ser um programa comercial, possui uma arquitetura fechada (*black-box*). Portanto, não é possível analisar como são realizadas as verificações, bem como, torna-se limitada a inserção de novas regras personalizadas.

- SMARTreview

Trata-se de um *add-on* integrado ao software Revit, que lida com checagem de regras de forma interativa e iterativa (REVIT ADD-ONS, 2017). Segundo a desenvolvedora, pode ser usada em diversas fases do projeto, podendo ser utilizada no início como auxílio na tomada de decisões ou em etapa de finalização, para determinação dos requerimentos que uma edificação irá necessitar, com vias em projetos de segurança, como o projeto de prevenção e combate à incêndio e pânico.

Ainda de acordo com a desenvolvedora, o programa possui diversos benefícios como um projeto que respeita as observâncias normativas, aprimora o conhecimento sobre regras e colaboração entre

equipe, aumenta a segurança e performance da edificação, reduz custos associados ao projeto, entre outros.

- FORNAX

Criado pela *Singapore CORENET (Construction and Real Estate Network)*, o FORNAX pode ser considerado como o primeiro esforço substancial no que se refere à checagem de regras de modelos BIM (NAWARI, 2013). Segundo o autor, o programa possui arquitetura base em C++ e expande o uso dos dados embutidos no IFC.

O governo de Singapura adota o software e descreve o seu funcionamento básico (CORENET, 2016). A submissão de modelos para checagem é realizada online. Na sequência, o sistema verifica o modelo de acordo com diversas regras pertinentes ao caso. Embutido no software, estão diversas normativas, incluindo regras relativas à construção de edificação, redesenvolvimento urbano, leis contra incêndio e pânico, de instalações de gás, entre outras. Ainda segundo o site oficial governamental, a plataforma CORENET já recebeu mais de 3.5 milhões de submissões online de projetos e desde 2001, trouxe uma redução de custos aos cofres públicos de Singapura, de 30 milhões de dólares singapurianos ou 113,7 milhões de reais.

- Jotne EDMmodelChecker

O EDMmodelChecker fornece validação de qualquer tipo de arquivo de dados baseados em EXPRESS, linguagem de modelagem de domínio específico, base dos arquivos IFC (NAWARI, 2013). Realiza uma verificação de consistência dos dados com as regras e restrições do esquema EXPRESS e faz um diagnóstico em níveis de instância e atributos.

Ele trabalha com base em composições de regras, em que é possível a definição complementar de diretrizes e restrições para esquemas EXPRESS específicos. Faz a verificação do modelo inteiro ou de entidades individuais, os conflitando com regras específicas (JOTNE EPM TECHNOLOGY NEWS, 2022).

- Dynamo

O Dynamo é um programa gratuito de linguagem visual de programação (DYNAMO BIM, 2021), que pode ser utilizado tanto de forma individual, sem a necessidade de outro software, ou como um *plug-in* para outros programas, como o Revit ou o Maya (DYNAMO PRIMER, 2022). É utilizado para acelerar diversos processos que, de outra forma, seriam feitos manualmente, obtendo assim, um ganho expressivo de tempo e produtividade (SHISHINA; SERGEEV, 2019).

Possui uma vasta biblioteca nativa, sendo sempre atualizada pela comunidade de usuários (DYNAMO PRIMER, 2022). Conta com ferramentas para desenvolvimento de geometrias complexas, definição de topologias específicas, integralização de espaço e ferramentas de localização por mapas (AUTODESK UNIVERSITY, 2022).

- Grasshopper

O Grasshopper é um programa de linguagem visual, que possui grande integração com o software de modelagem Rhinoceros 3D (Rhino's 3D) (GRASSHOPPER3D, 2022). Como todos os programas que lidam com linguagem visual de programação, não necessita de um amplo conhecimento de programação ou desenvolvimento de *scripts*.

Possui diversas funções voltadas para o desenvolvimento de geométricas complexas, sejam elas arquitetônicas ou voltadas para o design de itens diversos (RHINOCEROS, 2022a). Conta também com componentes que buscam um número grande de objetos e verificam se eles se interseccionam com outros elementos, ou seja, um detector de colisões (RHINOCEROS, 2022b).

- Marionette

O software Marionette é atribuído como o primeiro software de linguagem de programação, vinculado à um software BIM (ADMIN, 2016). Possui integração direta com o programa Vectorworks e também conta com uma extensa biblioteca de componentes (ALTIERI, 2020).

A Tabela 1 traz uma comparação entre os principais softwares de linguagem de programação.

Tabela 1: Comparação entre softwares VPL

Software	Dynamo	Grasshopper	Marionette
Linguagem de programação na qual o software foi desenvolvido?	C#	VB.Net e C#	Python
Atrelado a qual software?	Revit	Rhinoceros 3D	Vectorworks
Custo	Atrelado ao Revit. Versão <i>standalone</i> gratuita (<i>opensource</i>)	Atrelado ao Grasshopper	Atrelado ao Vectorworks
Pode ser usado em outros softwares?	Sim. Civil 3D, Alias Design, Formit, Advanced Steel e Maya	Archicad	Não
Possui versão <i>standalone</i> ?	Sim	Não	Não
Possui pacotes ou <i>add-ons</i> ?	Sim	Sim	Sim
Possibilidade de desenvolvimento próprio de pacotes ou <i>scripts</i> próprios?	Sim	Sim	Sim

Fonte: Autor

2.2 REVISÃO SISTEMÁTICA DA LITERATURA

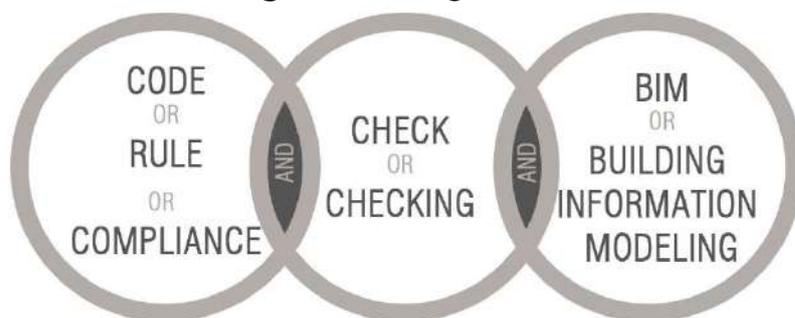
Vários são os métodos efetivos para a condução de uma Revisão Sistemática da Literatura (RSL) (MORANDI; CAMARGO, 2015). Todavia, é perceptível a existência um núcleo comum entre os diversos modos de se tratar uma RSL. Essa pesquisa é realizada através do Método Integrado, proposto por Morandi e Camargo, 2015, que compilam e ampliam os métodos existentes na literatura. Esse método passa pelas seguintes etapas: (i) definição do tema central e do *framework* conceitual; (ii) escolha da equipe de trabalho; (iii) estratégia de busca; (iv) busca, elegibilidade e codificação; (v) avaliação de qualidade; (vi) síntese dos resultados; (vii) apresentação do estudo.

Esta RSL possui como tema central a análise de pesquisas que envolvem a checagem automática de regras em BIM, com foco na verificação dos procedimentos adotados em casos de uso. A principal questão que motivou a RSL foi: em relação à checagem automática de regras em BIM, quais as principais ferramentas utilizadas, como se deu o uso do esquema IFC, quais os procedimentos e os desafios encontrados para desenvolvimento dos projetos.

Em relação a equipe de trabalho, essa RSL foi desenvolvida pelo autor dessa dissertação, com auxílio da orientadora.

No que se refere a estratégia de busca, foram utilizadas as bases de dados: *ScienceDirect*, *Scopus* e *Web of Science*, sendo que para cada uma dessas bases de dados, foram feitas buscas em artigos de periódicos, que estivessem em inglês, com ano de publicação entre 2016 e 2021, excluindo-se artigos de revisão. As *strings* de busca foram as seguintes: ((“CODE” OR “RULE” OR “COMPLIANCE”) AND (“CHECK” OR “CHECKING”) AND (“BIM” OR “BUILDING INFORMATION MODEL”)). Esses termos foram pesquisados no resumo, título ou palavra-chave dos artigos, Figura 7.

Figura 7 *Strings* de busca



Fonte: Autor

No que se refere à busca, elegibilidade e codificação, foram definidos critérios de exclusão para obtenção dos dados bibliométricos e consequente análise das pesquisas: (1) artigos duplicados e; (2) artigos em outro idioma que não o inglês. Como critérios de inclusão (3), foram considerados somente os artigos que tratavam sobre checagem automática de regras. Para tanto, foram lidos os títulos e os resumos.

Inicialmente, foi realizada uma análise bibliométrica com todos os artigos que passaram nos critérios de exclusão e inclusão. Os artigos foram classificados segundo o ano de publicação, periódico, coocorrência das palavras-chaves, nomenclatura.

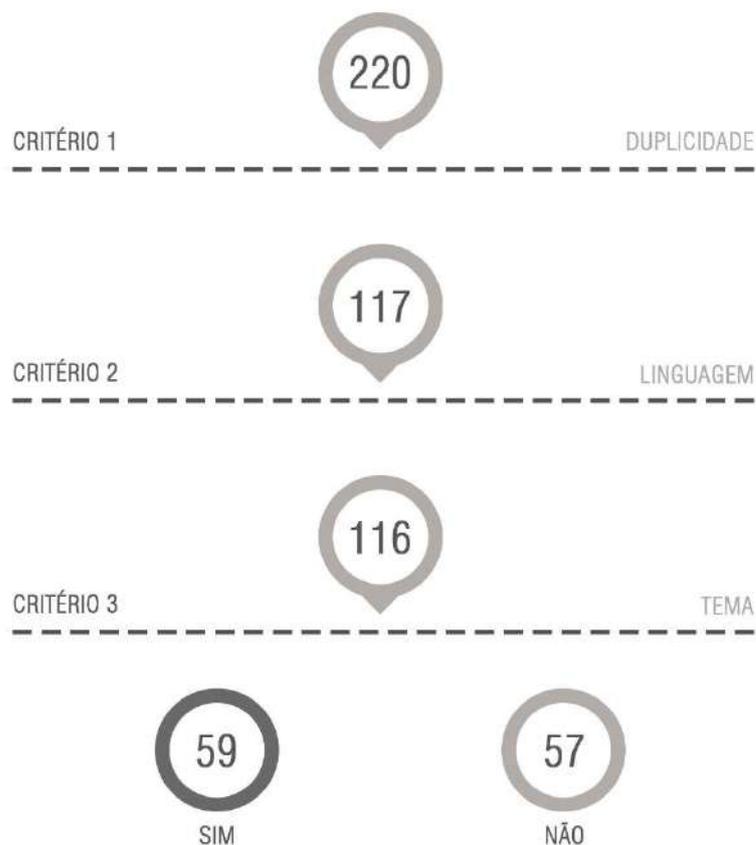
Em relação à avaliação da qualidade, foram selecionados estudos pertinentes para a revisão. Verificou-se sua adequação ao foco da revisão, ou seja, casos em que a checagem automática de regras tenha sido descrita no artigo. Para tanto, foi realizada uma segmentação segundo a pertinência do estudo, discriminando os demais focos de pesquisa abordados nos artigos.

Posteriormente, foi realizada a síntese dos resultados com os artigos que apresentaram um caso de uso. Tais artigos foram classificados de acordo com o propósito da checagem, a abordagem utilizada, os softwares de modelagem e de checagem utilizados, se fazem uso do esquema IFC, qual a linguagem de programação utilizada pelos autores, no caso de desenvolvimento de ferramentas específicas e dificuldades encontradas.

2.2.1 Busca, elegibilidade e codificação

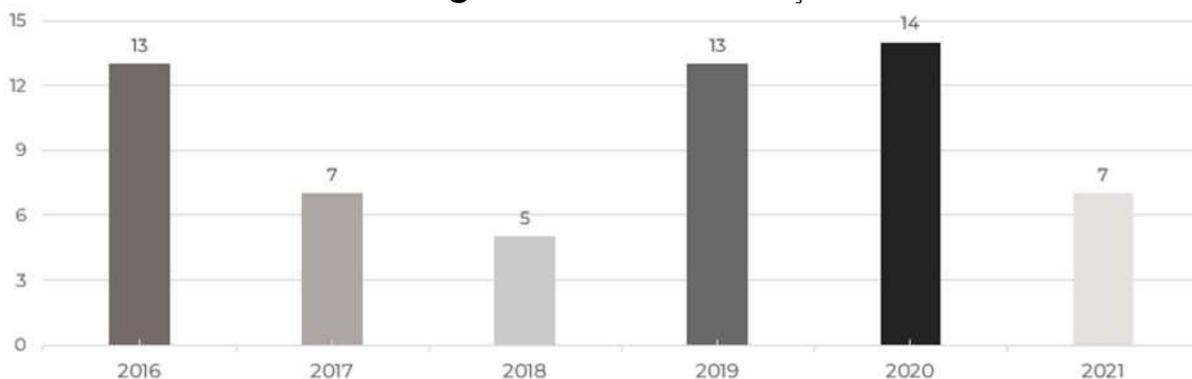
Após as pesquisas nas bases de dados, foram encontrados 52 artigos na *ScienceDirect*, 82 artigos na *Scopus* e 86 artigos na *Web of Science*, totalizando 220 artigos. Ao aplicar o critério de exclusão de duplicidade de artigos, foram obtidos 117 trabalhos. O critério de exclusão de idioma, resultou na remoção de um artigo.

Em relação ao critério de inclusão, após a leitura dos títulos e resumo, verificou-se que 57 não tratavam de checagem automática de regras e 59 tratavam do tema. Assim, os artigos aprovados corresponderam à 50,86% do total de artigos, Figura 8.

Figura 8 Artigos Aprovados

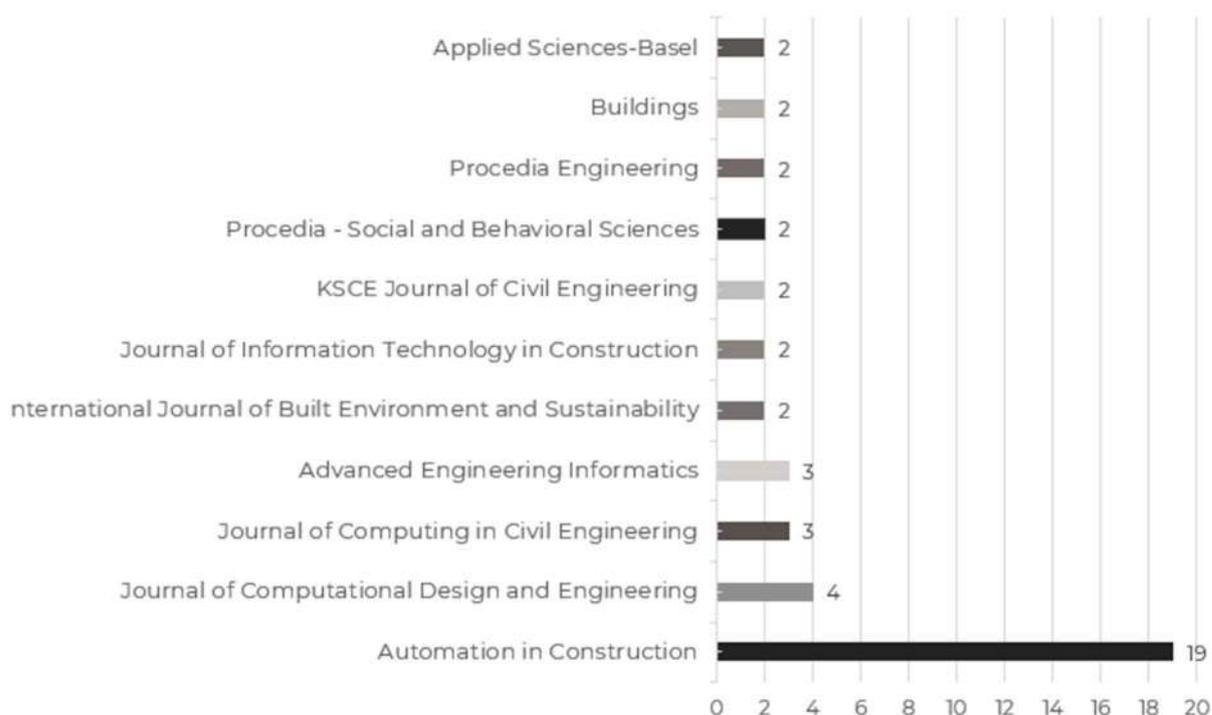
Fonte: Autor

A partir dos 59 artigos resultantes, foi realizada uma análise bibliométrica, visando obter uma visão geral sobre a produção dos artigos. Os resultados mostram 13 artigos publicados em 2016, sete publicados em 2017, cinco publicados em 2018, 13 publicados em 2019, 14 publicados em 2020 e sete publicados em 2021, Figura 9. Obteve-se uma média de 9,83 artigos publicados ao ano, no período indicado, sem, entretanto, haver um crescimento/decrescimento contínuo.

Figura 9 Ano de Publicação

Fonte: Autor

Também foi possível analisar, quais periódicos mais publicam sobre o tema. O periódico *Automation in Construction* é o principal, sendo que dentre os 59 artigos aprovados, 19 foram publicados nele, quatro na *Journal of Computational Design and Engineering*, três nas revistas *Journal Of Computing In Civil Engineering* e *Advanced Engineering Informatics*,. Os demais periódicos indicados, tiveram apenas duas publicações, Figura 10.

Figura 10 Publicações por Periódico

Fonte: Autor

Sobre a nomenclatura, foi possível constatar que a checagem de regras em BIM, por vezes é denominada como *automated code checking*, *automated code compliance checking*, *automated compliance audit*, *automated compliance checking*, mas percebe-se que na maioria dos casos, são utilizados termos genéricos como *rule cheking (RC)*, associada contextualmente a alguma especificidade da pesquisa sendo feita no momento, como “*automated safety rule checking*”.

Fica claro que não existe um consenso sobre a nomenclatura específica. Os termos mais utilizados foram: (a) *Rule Checking* - sete vezes; (b) *Automated Compliance Checking* - seis vezes; (c) *Automated Code Checking* - quatro vezes; (d) *Automated Code Compliance Checking* - quatro vezes; (e) *Automated Rule Checking* - quatro vezes; e (f) *Code Checking* - quatro vezes. Os demais nomes apareceram entre uma e duas vezes e estão demonstrados na Tabela 2.

Tabela 2: Nomenclatura adotada nas pesquisas

Nomenclatura	Quantidade de vezes que aparece	Fonte
Rule Checking	7	(PARK; LEE; LEE, 2016; PAUWELS et al., 2017; JI; LEITE, 2018; LEE; EASTMAN; SOLIHIN, 2018; FAN; CHI; PAN, 2019; SOMAN; MOLINA-SOLANA; WHYTE, 2020)
Automated Compliance Checking	6	(ZHANG; EL-GOHARY, 2016; ZHONG et al., 2020; ALTINTAS; ILAL, 2021; AMOR; DIMYADI, 2021; ZHOU; EL-GOHARY, 2021)
Automated Code Checking	4	(FAROOQ; SHARMA; KUMAR, 2018; HOSSAIN; AHMED, 2019; KIM et al., 2020; KINCELOVA et al., 2020)
Automated Code Compliance Checking	4	(SOLIHIN; EASTMAN, 2016; BLOCH; SACKS, 2018, 2020; NAWARI, 2019)
Automated Rule Checking	4	(GHANNAD et al., 2019; KIM et al., 2019; SCHWABE; TEIZER; KÖNIC, 2019; SONG et al., 2020)
Code Checking	4	(CIRIBINI; MASTROLEMBO VENTURA; PANERONI, 2016; TÜRKYILMAZ, 2016; CHOI; KIM, 2017; JIANG et al., 2019)

Nomenclatura	Quantidade de vezes que aparece	Fonte
Automated Model Checking	2	(BEAZLEY; HEFFERNAN; MCCARTHY, 2017; SYDORA; STROULIA, 2020)
Automated Safety Rule Checking	2	(TAKIM; ZULKIFLI; NAWAWI, 2016; ZULKIFLI; TAKIM; NAWAWI, 2016)
Automatic Rule Checking	2	(LEE et al., 2019; GUO; ONSTEIN; LA ROSA, 2021)
Compliance Checking	2	(ILAL; GUNAYDIN, 2017; SEGHIER; AHMAD; LIM, 2019)
Automated Compliance Audit	1	(DIMYADI et al., 2016)
Automated Regulatory Compliance Checking	1	(BEACH; HIPPOLYTE; REZGUI, 2020)
Automated Rule Base Inspection	1	(YUAN et al., 2019)
Automated Rule Based	1	(LEE; SOLIHIN; EASTMAN, 2019)
Automated Rule Compliance	1	(KHAN et al., 2019)
Automated Rule Compliance Checking	1	(SOLIMAN-JUNIOR; FORMOSO; TZORTZOPOULOS, 2020)
Automated Rule-Based Checking System	1	(YING; LEE, 2021)
Automatic Checking	1	(WARREN; SALEEB, 2020)
Automatic Checking Of Semantic Information	1	(BURGGRAF et al., 2021)
Code Checking System	1	(LEE; CHIANG, 2016)
Code Compliance Checking	1	(SHAHI; MCCABE; SHAHI, 2019)
Rule Based	1	(KANG, 2020)
Rule Based Checking	1	(LAI et al., 2020)
Rule Based Code Checking	1	(GETULI et al., 2017)
Rule-Based Checking	1	(LEE et al., 2016b)
Utility Compliance Checking	1	(LI; CAI; KAMAT, 2016)

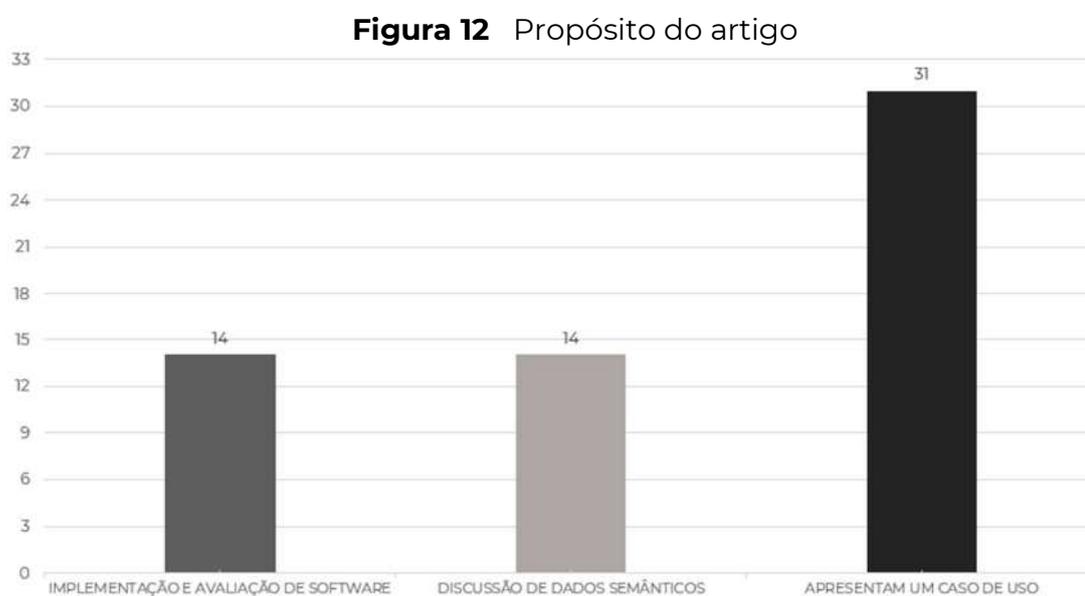
Fonte: Autor

2.2.2 Avaliação da qualidade

A avaliação da qualidade foi realizada com todos os artigos aprovados no processo de busca, elegibilidade e codificação. Buscando classificar a relevância das pesquisas para o estudo em questão, os artigos foram segmentados segundo seu principal propósito, quais sejam: (a) artigos que tratam da implementação e avaliação de um software ou *framework* de checagem de regras; (b) artigos que lidam com a questão

semântica de informações de modelos, objetos e processos BIM e; (c) artigos que utilizam BIM para checagem automática de regras.

Foi possível verificar que 14 artigos tratam sobre os dados semânticos, 14 artigos trazem informações sobre implementação e 31 artigos apresentam um caso de aplicação de alguma sistemática de checagem de regras, Figura 12. Existe um equilíbrio entre trabalhos que tratam de implementação e validação de software e discussão de dados semânticos e uma maior quantidade de pesquisas apresentam um caso de uso para uma disciplina específica do setor AEC.



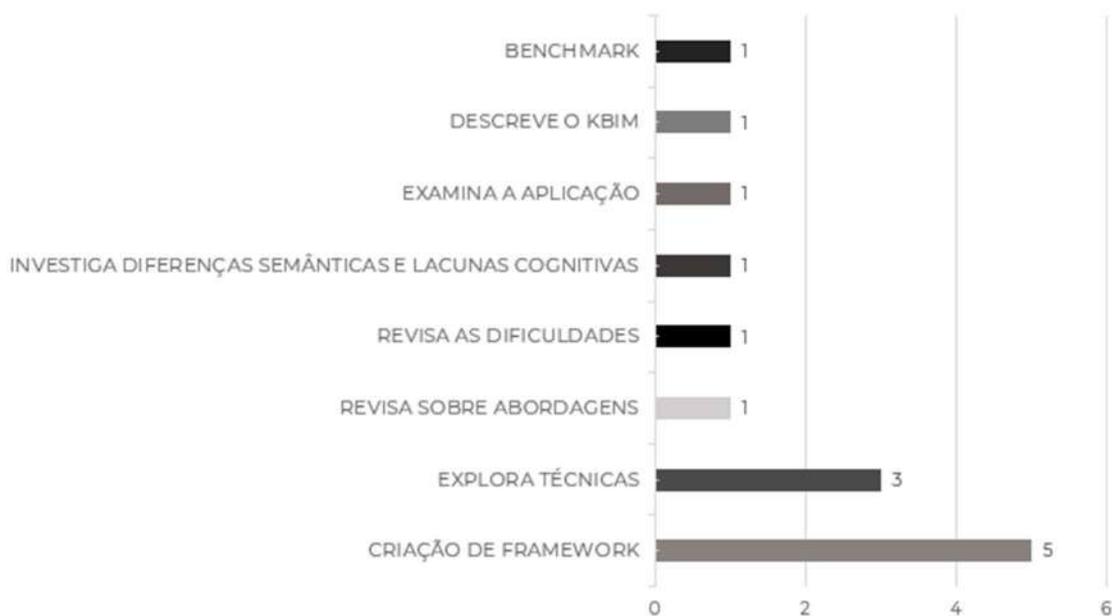
Fonte: Autor

Os artigos que tratam da **implementação e avaliação de software**, falam sobre testes de (a) *benchmark* – 1 artigo (PAUWELS et al., 2017); (b) descrevem o KBIM – 1 artigo (PARK; LEE; LEE, 2016); (c) examina a aplicação da checagem de regras para ganho de produtividade – 1 artigo (WARREN; SALEEB, 2020); (d) investiga as diferenças semânticas e lacuna cognitiva entre os participantes do processo de checagem – 1 artigo (LEE; CHIANG, 2016); (e) revisa sobre as dificuldades de aplicação do sistema de checagem de regras – 1 artigo (BEACH; HIPPOLYTE; REZGUI, 2020); (f) revisa sobre as abordagens tomadas até o momento para implementação de sistemáticas de checagem automática de regras – 1 artigo (AMOR; DIMYADI,

2021) ;(g) explora técnicas, isto é, explora possíveis técnicas de aplicação para desenvolvimento do uso de checagem automática de regras – 3 artigos (TAKIM; ZULKIFLI; NAWAWI, 2016; BEAZLEY; HEFFERNAN; MCCARTHY, 2017; PREIDEL; DAUM; BORRMANN, 2017) e; (h) cria um *framework* para utilização da checagem automática de regras – 5 artigos (ZULKIFLI; TAKIM; NAWAWI, 2016; SHAHI; MCCABE; SHAHI, 2019; SOLIMAN-JUNIOR; FORMOSO; T

ZORTZOPOULOS, 2020; ZHONG et al., 2020; GUO; ONSTEIN; LA ROSA, 2021) – Figura 13. Sobre os trabalhos que trataram de implementação e avaliação de software, a grande maioria dele possui foco na criação de um *framework* para checagem automática de regras de alguma disciplina específica do setor de AEC. Outros trabalhos fazem testes em softwares distintos, para verificação do tempo de processamento e examinam a aplicação para o ganho de produtividade que o setor teria com a implantação de tais sistemáticas.

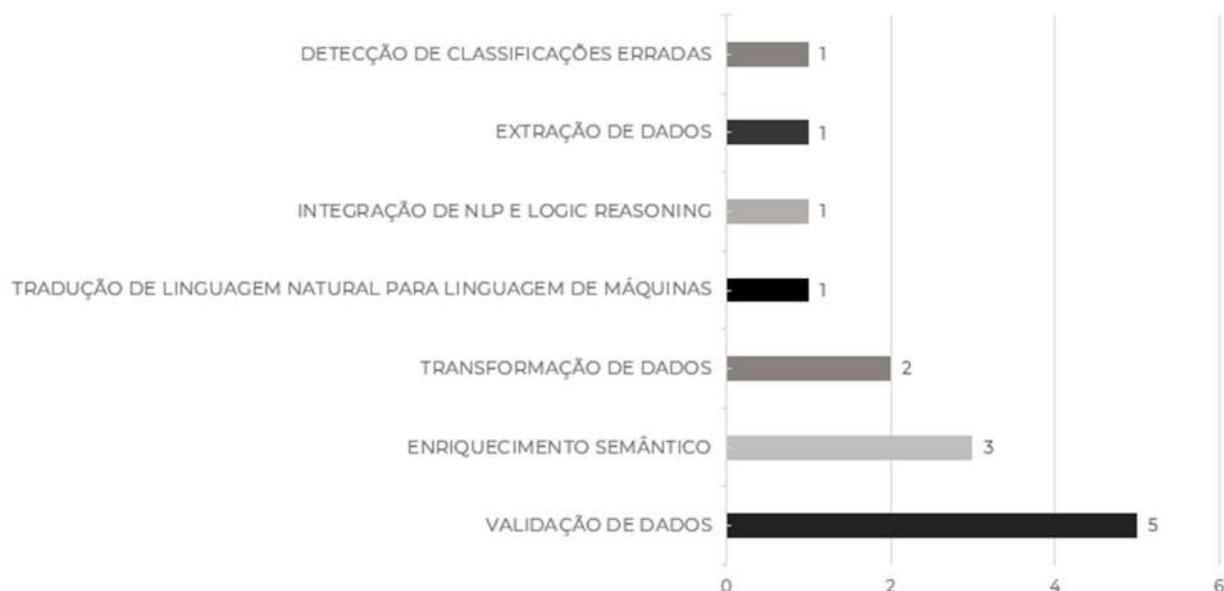
Figura 13 Implementação e Avaliação de *Software*



Fonte: Autor

Já os artigos que tratam de **discussão de dados semânticos**, foram categorizados da seguinte forma: (a) detecção de classificações erradas, em que é proposto um sistema em *machine learning* para detectar potenciais classificações erradas que possam ocorrer o mapeamento de um

certo conjunto de dados – 1 artigo (KOO; SHIN, 2018); (b) extração de dados, no qual são utilizadas técnicas de extração de dados, para tradução de sentenças, fazendo uso do *deep learning* – 1 artigo (SONG et al., 2020); (c) integração de *Natural Language Process (NLP)* e *Logic Reasoning*, em que é feita uma integração de duas técnicas para lidar com dados semânticos – 1 artigo (ZHANG; EL-GOHARY, 2017); (d) tradução de linguagem natural (LN) para linguagem de máquinas (LM) – 1 artigo (LEE et al., 2016a); (e) transformação de dados, na qual são realizados alinhamento de dados com o uso do *deep learning* e dados são transformados em esquemas simplificados – 2 artigos (SOLIHIN et al., 2020; ZHOU; EL-GOHARY, 2021); (f) enriquecimento semântico, em que são comparados métodos distintos de enriquecimento semântico – 3 artigos (ZHANG; EL-GOHARY, 2016; BLOCH; SACKS, 2018, 2020)– e; (g) validação de dados – 5 artigos (LEE et al., 2016b; LEE; EASTMAN; SOLIHIN, 2018; LEE; SOLIHIN; EASTMAN, 2019; SOMAN; MOLINA-SOLANA; WHYTE, 2020; YOUNG-CHEOL; EASTMAN; SOLIHIN, 2021), Figura 14. É perceptível que o tratamento de dados semânticos é indispensável, especialmente para a automatização da tradução de regras e normativas, processo esse que muitas das vezes são inseridos de forma manual. Contudo, tais processos são complexos, envolvendo técnicas de *machine learning*, *deep learning*, *natural language process*, entre outros. Devida atenção deve ser dada à validação dos dados que compõe a checagem automática, pois uma atribuição errônea pode vir a gerar resultados falso-positivos, diminuindo o grau de eficácia dos sistemas.

Figura 14 Discussão de Dados Semânticos

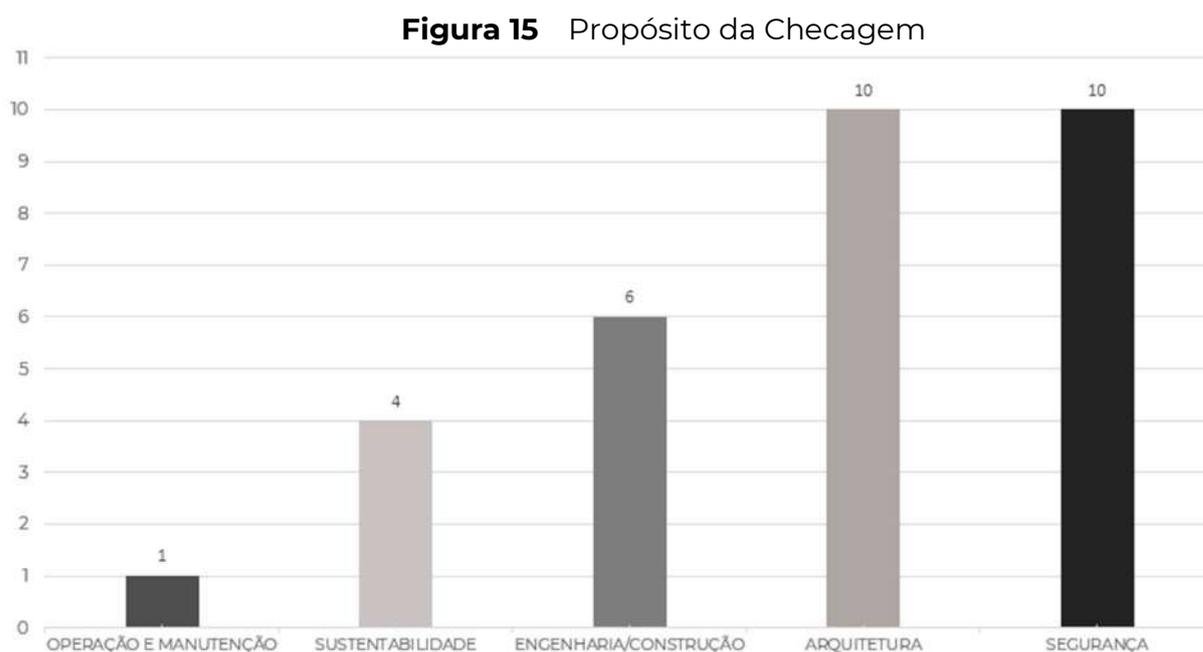
Fonte: Autor

2.2.3 Síntese dos resultados

Os 31 artigos que **apresentaram um caso de uso** foram estudados, a fim de se responder as questões que motivaram a RSL. Assim, os artigos foram classificados de acordo com: (i) propósito da checagem; (ii) abordagem da checagem de regras; (iii) se os trabalhos utilizaram ou não o arquivo IFC; (iv) quais os principais softwares de modelagem utilizados; (v) quais os softwares específicos para checagem de regras; (vi) quais as linguagens de programação mais utilizadas, para trabalhos que desenvolveram seus próprios sistemas e; (vii) quais as dificuldades encontradas para realização da checagem automática de regras.

De acordo com o propósito da checagem, tem-se: (a) manutenção e operação – 1 artigo; (b) sustentabilidade, com propostas de checagem automática de regras, voltadas para checagem e subsequente aprovação de modelos para obtenção de certificados e selos ambientais – 4 artigos; (c) engenharia/construção, que são artigos que lidam com temas correlatos à engenharia ou construção de um modo geral, como planejamento de canteiro de obras, estimativa de custos para projetos, entre outros – 6 artigos; (d) segurança, que são artigos que tratam de checagens voltadas

para segurança de canteiro de obras, prevenção de acidentes, segurança contra quedas, prevenção, segurança e combate ao incêndio e pânico, entre outros – 10 artigos – e; (e) arquitetura, como artigos que tratam de acessibilidade, aprovação de projetos, seguindo alguma sistemática específica – 10 artigos, Figura 15, Tabela 3.



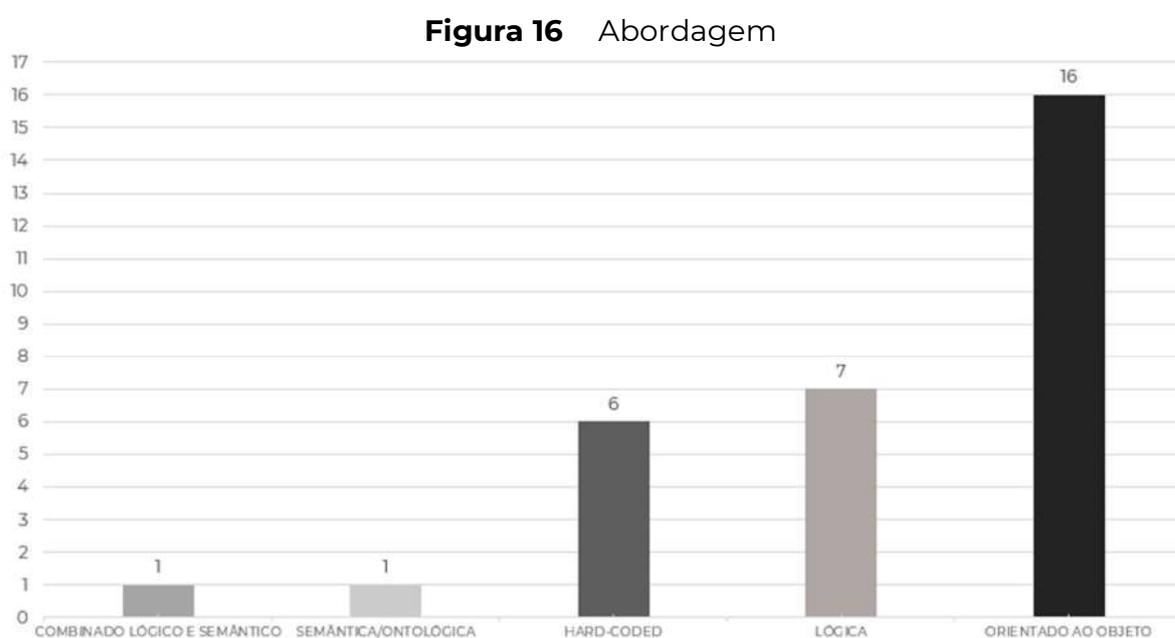
Fonte: Autor

Tabela 3: Propósito da checagem e suas fontes

Propósito da checagem	Fonte
Manutenção e operação	(AKANMU; OLAYIWOLA; ALFRED OLATUNJI, 2020)
Sustentabilidade	(CHOI; KIM, 2017; SEGHIER et al., 2017; JIANG et al., 2019; KANG, 2020)
Engenharia e construção	(LI; CAI; KAMAT, 2016; FAROOQ; SHARMA; KUMAR, 2018; JI; LEITE, 2018; SCHWABE; TEIZER; KÖNIG, 2019; SEGHIER; AHMAD; LIM, 2019; BURGGRÄF et al., 2021)
Segurança	(CIRIBINI; MASTROLEMBO VENTURA; PANERONI, 2016; DIMYADI et al., 2016; HONGLING et al., 2016; GETULI et al., 2017; HOSSAIN; AHMED, 2019; KHAN et al., 2019; KIM et al., 2019; LEE et al., 2019; YUAN et al., 2019; KINCELOVA et al., 2020)
Arquitetura	(TÜRKYILMAZ, 2016; ILAL; GUNAYDIN, 2017; FAN; CHI; PAN, 2019; GHANNAD et al., 2019; NAWARI, 2019; KIM et al., 2020; LAI et al., 2020; SYDORA; STROULIA, 2020; ALTINTAS; ILAL, 2021; YING; LEE, 2021)

Fonte: Autor

Em relação à checagem automática de regras, observa-se que a abordagem orientada ao objeto foi a mais utilizada, com 16 artigos. Na sequência, temos a abordagem lógica, com sete artigos, a abordagem *hard-coded* com seis e a abordagem semântica/ontológica e um combinado entre as abordagens lógica e semântica com um artigo, Figura 16, Tabela 4. Poucos trabalhos lidam com a abordagem semântica/ontológica, pois trata-se de uma sistemática que envolve diversos processos e softwares específicos.



Fonte: Autor

Tabela 4: Abordagem e suas fontes

Abordagem	Fonte
Combinado lógica e semântica	(JIANG et al., 2019)
Semântica / Ontológica	(LEE et al., 2019)
<i>Hard-coded</i>	(TÜRKYILMAZ, 2016; CHOI; KIM, 2017; GETULI et al., 2017; JI; LEITE, 2018; HOSSAIN; AHMED, 2019; KIM et al., 2020)
Lógica	(CIRIBINI; MASTROLEMBO VENTURA; PANERONI, 2016; LI; CAI; KAMAT, 2016; ILAL; GUNAYDIN, 2017; FAN; CHI; PAN, 2019; YUAN et al., 2019; LAI et al., 2020; SYDORA; STROULIA, 2020)
Orientado ao objeto	(DIMYADI et al., 2016; HONGLING et al., 2016; SEGHIER et al., 2017; FAROOQ; SHARMA; KUMAR, 2018; NAWARI, 2019; SCHWABE; TEIZER; KÖNIG, 2019; SEGHIER; AHMAD; LIM, 2019; GHANNAD et al., 2019; KHAN et al., 2019;

Abordagem	Fonte
	KIM et al., 2019; AKANMU; OLAYIWOLA; ALFRED OLATUNJI, 2020; KINCELOVA et al., 2020; KANG, 2020; ALTINTAS; ILAL, 2021; YING; LEE, 2021; BURGGRÄF et al., 2021)

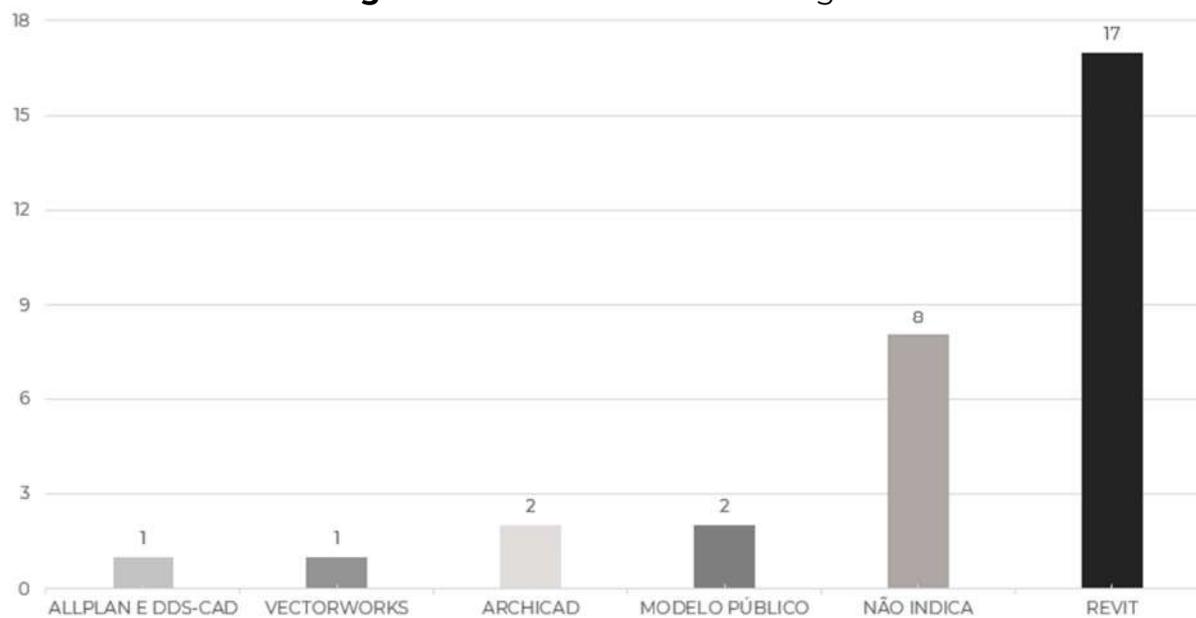
Fonte: Autor

A abordagem *hard-coded* acaba possuindo como limitações o fato de que, em geral, os softwares possuem uma arquitetura fechada (*black-box*), a qual o usuário não consegue aferir todas as etapas e processos de checagem. Atrelado à essa questão, tem-se os altos custos dos softwares.

Já a abordagem lógica possui talvez, a maior capacidade de automatização de todas as etapas do processo de checagem de regras, isso inclui a transcrição ou tradução de regras escritas em linguagem natural para linguagem de máquinas. No entanto, é uma técnica que exige um grande conhecimento sobre manipulação de dados, *machine learning*, *deep learning*, *natural language processing*, entre outros e isso acaba limitando o avanço dessa abordagem.

Possivelmente por oferecer uma facilidade na elaboração dos processos, a abordagem orientada ao objeto foi a mais utilizada nas pesquisas analisadas. Além do mais, nessa abordagem, os processos podem ser criados diretamente nos softwares proprietários, com a integração de programas de linguagem visual.

Com relação aos programas de modelagem, foi possível observar que a grande maioria – 17 artigos – utilizam o software Autodesk Revit para modelagem, oito não indicam qual software utilizaram, dois fazem uso de modelos públicos, disponíveis na internet, dois utilizam o Archicad, um utiliza o Vectorworks e um o AllPlan combinado com o DDS CAD, Figura 17. A grande maioria dos trabalhos não demonstra qual a versão do software foi utilizada.

Figura 17 Softwares de Modelagem

Fonte: Autor

Comumente, a interoperabilidade em BIM é realizada por meio do esquema IFC e por isso, foi analisado quais pesquisas o utilizaram em seus processos. Para todos os artigos que apresentaram caso de uso concreto, 18 artigos utilizaram o IFC nos seus processos de verificação de regras, enquanto 13 artigos não. Observou-se que a maioria dos trabalhos não traz uma distinção específica entre os esquemas de IFC, como o IFC 2x3 *Coordination View 2.0* e o IFC 4. Dentre os artigos que apresentam caso de uso e que fazem uso do arquivo IFC, é possível perceber uma preponderância com relação ao uso de softwares desenvolvidos pelos autores, Tabela 5. O uso de softwares comerciais também é feito pelos autores, como é o caso do Solibri, Drolls e Open IFC Tools e softwares comerciais exclusivos para uma determinada região, como é o caso do KBIM Assess e KBIM Assess-Lite.

Tabela 5: Softwares utilizados e suas fontes

Softwares utilizados	Fonte
Desenvolvimento próprio	(DIMYADI et al., 2016; HONGLING et al., 2016; ILAL; GUNAYDIN, 2017; JIANG et al., 2019; LEE et al., 2019; NAWARI, 2019; LAI et al., 2020; SYDORA; STROULIA, 2020; ALTINTAS; ILAL, 2021; YING; LEE, 2021)

Softwares utilizados	Fonte
Solibri	(CIRIBINI; MASTROLEMBO VENTURA; PANERONI, 2016; TÜRKYILMAZ, 2016; CHOI; KIM, 2017; GETULI et al., 2017; JI; LEITE, 2018; HOSSAIN; AHMED, 2019)
Drools e Open IFC Tools	(SCHWABE; TEIZER; KÖNIG, 2019)
KBIM Assess e KBIM Assess-Lite	(KIM et al., 2020)

Fonte: Autor

Com relação aos trabalhos que elaboraram sistemas próprios, temos que três deles o fizeram com base na linguagem de programação Visual Basic C#, três não indicam qual linguagem foi utilizada, dois utilizaram o Java, um utilizou o mvdXML e um o Visual C++, Figura 18.

Figura 18 Uso do IFC



Fonte: Autor

Com relação aos softwares utilizados para os procedimentos de checagem, independentemente do uso dos arquivos IFC, foi possível

verificar que quatro utilizaram o programa de linguagem visual Dynamo, seis utilizaram o Solibri Model Checker e 15 artigos fizeram a checagem de regras a partir do desenvolvimento próprio e específico de seu sistema, Figura 19. Os demais seis artigos utilizaram outros softwares, indicados na Tabela 6.

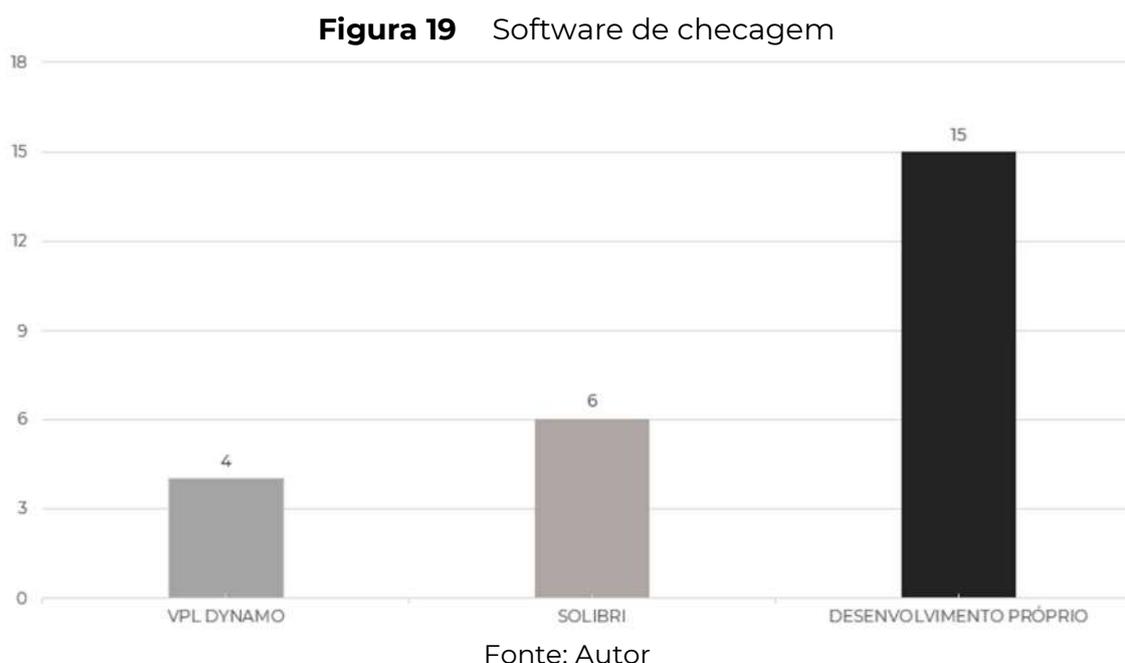


Tabela 6: Softwares de checagem de regras

Software de checagem	Fonte
Desenvolvimento Próprio	(LI; CAI; KAMAT, 2016; DIMYADI et al., 2016; ZHANG; EL-GOHARY, 2017; ILAL; GUNAYDIN, 2017; FAROOQ; SHARMA; KUMAR, 2018; NAWARI, 2019; YUAN et al., 2019; FAN; CHI; PAN, 2019; JIANG et al., 2019; LEE et al., 2019; AKANMU; OLAYIWOLA; ALFRED OLATUNJI, 2020; SYDORA; STROULIA, 2020; KANG, 2020; LAI et al., 2020; ALTINTAS; ILAL, 2021; YING; LEE, 2021)
Solibri	(CIRIBINI; MASTROLEMBO VENTURA; PANERONI, 2016; TÜRKYILMAZ, 2016; CHOI; KIM, 2017; GETULI et al., 2017; JI; LEITE, 2018; HOSSAIN; AHMED, 2019)
VPL Dynamo	(SEGHIER et al., 2017; SEGHIER; AHMAD; LIM, 2019; KINCELOVA et al., 2020; BURGGRÄF et al., 2021)
VPL Marionette	(GHANNAD et al., 2019)
VPL Grasshoper	(KHAN et al., 2019)
Unity 3d	(HONGLING et al., 2016)
KBVL	(KIM et al., 2019)
KBIM Assess e KBIM Assess-Lite	(KIM et al., 2020)
Drools + Open IFC Tools	(SCHWABE; TEIZER; KÖNIG, 2019)

Fonte: Autor

Em relação as dificuldades e limitações encontradas nos processos de verificação automática de regras, foram identificadas questões ligadas somente à tecnologia, sendo elas: (i) tratamento de dados; (ii) geometria; (iii) abrangência da investigação e; (iv) testes, Figura 20, Tabela 7.

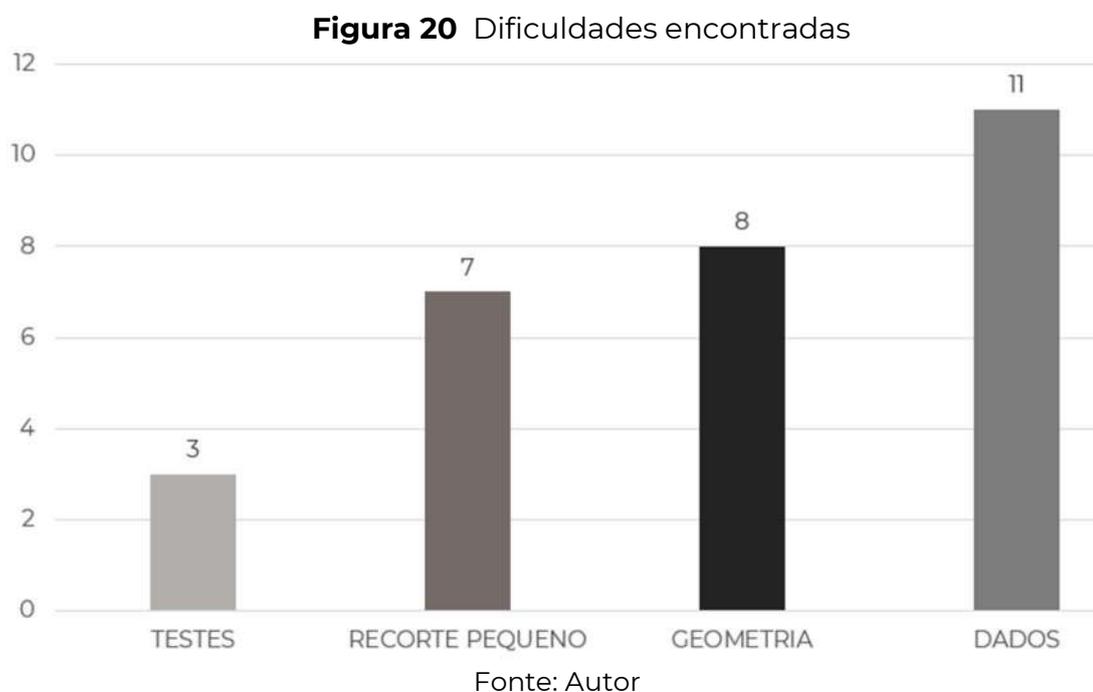


Tabela 7: Dificuldades encontradas e suas fontes

Dificuldades	Fonte
Testes	(DIMYADI et al., 2016; GETULI et al., 2017; KANG, 2020)
Abrangência da investigação	(CHOI; KIM, 2017; ILAL; GUNAYDIN, 2017; HOSSAIN; AHMED, 2019; KHAN et al., 2019; KIM et al., 2019, 2020; KINCELOVA et al., 2020)
Geometria	(JI; LEITE, 2018; FAN; CHI; PAN, 2019; NAWARI, 2019; SCHWABE; TEIZER; KÖNIG, 2019; SEGHIER; AHMAD; LIM, 2019; SYDORA; STROULIA, 2020; BURGGRÄF et al., 2021; YING; LEE, 2021)
Tratamento de dados	(HONGLING et al., 2016; LI; CAI; KAMAT, 2016; SEGHIER et al., 2017; FAROOQ; SHARMA; KUMAR, 2018; YUAN et al., 2019; GHANNAD et al., 2019; JIANG et al., 2019; LEE et al., 2019; AKANMU; OLAYIWOLA; ALFRED OLATUNJI, 2020; LAI et al., 2020; ALTINTAS; ILAL, 2021)

Fonte: Autor

2.2.4 Considerações

A RSL apresentada, teve como foco a análise de pesquisas que envolvem a checagem automática de regras em BIM. Esse é um processo com alto grau de complexidade, especialmente no que se refere à determinadas etapas de sua implementação. Percebe-se que as possibilidades de uso se estendem para diversos setores que possuem algum tipo de checagem manual de processos.

Foi possível observar que existem diversos trabalhos que apresentam estratégias diferentes para checagem automática de regras. Em várias situações, foram apresentados desenvolvimentos de aplicações específicas para os usos desejados. Isso se deve à dificuldade em alguns processos de interpretação de regras e estruturação lógica, especialmente no que se refere à transcrição da linguagem natural para a linguagem de máquina (FAROOQ; SHARMA; KUMAR, 2018; HOSSAIN; AHMED, 2019; NAWARI, 2019; SCHWABE; TEIZER; KÖNIG, 2019) e na criação de regras customizadas dentro de sistemas comerciais (SCHWABE; TEIZER; KÖNIG, 2019; SOLIMAN-JUNIOR; FORMOSO; TZORTZOPOULOS, 2020; BURGGRÄF et al., 2021).

O uso de mecanismos de *hard-coded* facilita o desenvolvimento de protocolos específicos, quando são utilizadas funções pré-programadas, gerando uma economia expressiva de tempo. Contudo, tais mecanismos acabam por ter uma versatilidade limitada, uma vez que não são editáveis na grande maioria das vezes. Assim, nessa situação, o usuário entra com os dados, ativa o mecanismo de checagem e recebe os resultados, porém, não é capaz de entender ou realizar modificações nas regras, somente nos valores dos parâmetros.

Algumas estratégias foram de implementação de linguagem visual de programação com a definição de modelos pré-configurados; a criação de *Application Program Interfaces* (API) auxiliares para programas proprietários e para software comercial de checagem; desenvolvimento de estruturação lógica, com a tradução de regras ou normas sendo feita de

forma automatizada, via RASE, PAS e combinações entre abordagens; a utilização de sistemáticas de *machine learning* e/ou *deep learning* para tradução de regras e para verificação de consistência de informações. A linguagem visual de programação diminui a necessidade do conhecimento de programações mais avançadas, pela sua própria conformação intuitiva, tornando-a assim, mais atrativa ao uso em geral.

Em relação às etapas propostas para a realização da checagem automática de regras, observou-se, em alguns trabalhos, que nem todas foram automatizadas (EASTMAN et al., 2009), como por exemplo na etapa de interpretação de regras e estruturação lógica, a transposição de regras em linguagem natural para linguagem de máquinas foi realizada de forma manual e na etapa de exportação dos resultados de checagem, não foi gerado automaticamente o relatório indicando os itens que passaram ou não nas verificações.

As análises realizadas contribuem de forma a entender como andam o desenvolvimento de sistemáticas de checagem automática de regras. Observou-se que não há uma abordagem unificada ou totalmente editável e interoperável, devido às complexidades envolvidas e demandas específicas. Entretanto, no que se refere à escolha da melhor abordagem a ser seguida, ou até mesmo quais processos automatizar, a discussão apresentada auxilia na escolha de sistemáticas mais eficientes para futuros trabalhos, minimizando o tempo consumido em estudos exploratórios.

Foi possível perceber diversos desafios nos procedimentos de criação de mecanismos que possibilitem a checagem automatizada de regras de modelos BIM. Questões relacionadas ao tratamento de dados foram as principais barreiras encontradas, incluindo dificuldades com a inserção de dados, inflexibilidade com relação às regras inseridas, identificação de informações conflitantes, fluxo de informações não automatizado.

Várias dificuldades estavam ligadas à geometria, sendo problemas no tratamento de configurações complexas (como curvas e objetos convexos), o não suporte a determinadas tipologias de objetos (como

rampas), necessidade de simplificação da modelagem ou de configurações específicas do modelo seguindo determinado padrão e falta de informações nos componentes BIM. Outra limitação significativa estava ligada à abrangência da investigação, nos casos em que foram explorados somente pontos específicos da normativa utilizada como base, a título de validação da ferramenta.

3 MATERIAIS E MÉTODOS

3.1 MÉTODO

Com o intuito de se desenvolver uma pesquisa orientada ao desenvolvimento de um artefato novo, que possua foco na mudança e com o objetivo de solucionar um problema existente, a pesquisa pretendida será conduzida nos moldes da *Design Science Research*.

Tal artefato pode ser ilustrado como uma representação simbólica ou uma instanciação física dos conceitos de *design* (GILL; HEVNER, 2013). O próprio processo de *design*, nesse caso tido como projeto, possui diversas camadas que estão relacionadas diretamente com o artefato. Dentre essas camadas para a elaboração do artefato, destacam-se: (i) espaço do *design*, (ii) camadas do artefato em construção, subdivida em quatro etapas e (iii) uso do artefato, Figura 21.



Fonte: Autor

A primeira etapa para a elaboração do artefato é aquela que possui o conjunto de soluções possíveis para resolver o problema previamente identificado. Nela são propostos os possíveis caminhos para o desenvolvimento do artefato. Tendo uma solução adequada sido escolhida, deve-se passar para as camadas subsequentes e analisar a sua viabilidade e utilidade, para a partir de então, proceder com a sua representação e construção (DRESCH; LACERDA; ANTUNES JÚNIOR, 2015).

Com os conceitos principais do artefato definidos, avança-se para sua caracterização, sendo que eles podem ser classificados como: (i) *Constructo*, que são as codificações ou vocabulários de um domínio e são utilizados para descrever os problemas e propor soluções; (ii) *Modelo*, que são os conjuntos de regras que estabelecem conexões entre os constructos; (iii) *Método*, que são todas as etapas declaradas para conclusão de determinada tarefa; e (iv) *Instanciação*, que são artefatos que operacionalizam outros artefatos e demonstram ou não a sua viabilidade e acurácia, podendo informar como utilizar ou implementar outro artefato (MARCH; SMITH, 1995).

A pesquisa proposta será do tipo exploratória e o artefato proposto será do tipo Método e Instanciação, uma vez que o mesmo descreve etapas para a conclusão de determinada tarefa, em que num sistema busca a sua melhoria e otimização e também serão previstos os possíveis resultados de sua implementação no ambiente real.

Essa pesquisa será desenvolvida em seis etapas, Figura 22, quais sejam:

01: Identificação do problema a ser estudado, no qual é visualizado que no Setor de Planejamento e Desenvolvimento Urbano de Poços de Caldas, existe uma enorme burocracia com relação à aprovação de projetos arquitetônicos / construção e está instaurado um processo que leva à um gasto substancial de tempo e esforço por parte dos projetistas, bem como dos analistas de projeto – Item 1 INTRODUÇÃO.

02: Conscientização do problema e RSL, no qual é buscado o máximo de informações pertinentes, relacionados ao problema, com o intuito de compreender de forma completa, todas as suas faces e com consultas às bases de conhecimento, captar conhecimentos prévios acerca do assunto tratado, o que auxiliará em sua resolução - Item 2 FUNDAMENTAÇÃO TEÓRICA e 2.2 REVISÃO SISTEMÁTICA DA LITERATURA.

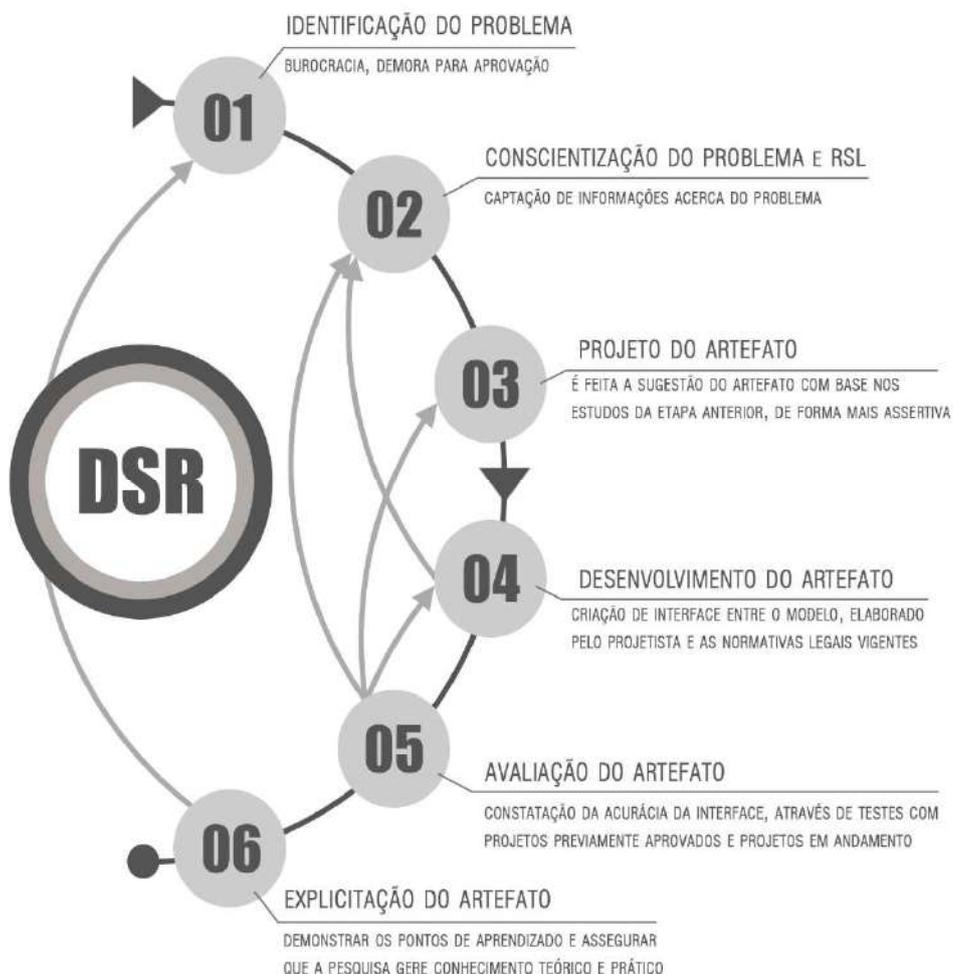
03: Projeto do artefato, no qual com base nos estudos realizados em etapa anterior é feita a seleção do objeto de estudo e as premissas iniciais são estudadas – Item 4 PROJETO DO ARTEFATO.

04: Desenvolvimento do artefato, com a criação de interface entre o modelo elaborado pelo projetista e as normativas legais vigentes, tendo em vista a verificação de normas, baseadas num conjunto de regras estabelecidas pela Lei de Uso e Ocupação do Solo (LUOS), da cidade de Poços de Caldas, MG, em que a interface irá realizar a leitura do modelo e indicará se o projeto está passível ou não de aprovação, dentro dos parâmetros selecionados – Item 5 DESENVOLVIMENTO DO ARTEFATO.

05: Avaliação do artefato consiste na verificação da confiabilidade da resposta do sistema em relação aos parâmetros urbanísticos, como afastamentos laterais e de fundo, recuo frontal, altura máxima permitida, área construída, área permeável, taxa de ocupação, coeficiente de aproveitamento e taxa de permeabilidade – Item 6 AVALIAÇÕES.

06: Explicitação do aprendizado, para assegurar que a pesquisa tenha continuidade e que possa servir de subsídio para a geração de conhecimento, tanto teórico como prático – Item 7 DISCUSSÃO E CONCLUSÃO

Figura 22 Fluxograma das etapas da pesquisa



Fonte: Autor

3.2 MATERIAIS

Para o desenvolvimento desta pesquisa, foram utilizados os seguintes programas e componentes:

Computador – computador próprio, com processador AMD Ryzen 5 3600, 16 GB de memória RAM (*Random Access Memory*), placa de vídeo dedicada Radeon RX 550, SSD (*Solid State Drive*) de 500 GB de armazenamento e sistema operacional Windows 10.

Autodesk Revit 2023 (v. 23.0.11.19) – como programa base para desenvolvimento do modelo a ser testado.

Dynamo (v. 2.13.1.3891) – como programa de linguagem visual, para desenvolvimento da lógica de programação, com entradas automáticas e manuais em Python.

4 PROJETO DO ARTEFATO

Neste capítulo, é realizada a proposição do artefato, definido o objeto do estudo e são desenvolvidos estudos de verificação de modelos. De forma inicial, pretende-se trabalhar com um recorte mais simplificado em termos de legislação, para uma avaliação do artefato e das dificuldades de seu desenvolvimento.

4.1 PROPOSIÇÃO DO ARTEFATO

A título de experimentação, o artefato proposto, visa verificar automaticamente critérios objetivos da legislação, como os recuos frontais e afastamentos laterais e de fundos, de uma edificação unifamiliar. O intuito é observar, por meio de seu desenvolvimento, os desafios tecnológicos encontrados, buscando uma maior compreensão desses. O artefato é proposto para que projetistas possam utilizá-lo em uma pré-aprovação digital de projetos que seriam submetidos em fase posterior à prefeitura.

Conforme apontado na RSL, os softwares do tipo VPL oferecem maior flexibilidade para criação de regras específicas, facilidade de aprendizado e uso dessa linguagem de programação, disponibilização de componentes pré-programados, além de possuir uma interface com o usuário intuitiva. Corroborando, os três softwares apontados na Tabela 1: são gratuitos. Desses, o Dynamo foi o mais utilizado nas pesquisas levantadas pela RSL (Tabela 6:). Adicionalmente, o software de modelagem mais utilizado nesse contexto é o Revit, (Figura 17).

Portanto, o sistema foi desenvolvido utilizando o software Dynamo, em sua versão 2.13.1.3891 e testado com o software Revit 2023, em sua versão 23.0.11.19. Dentro da interface do próprio Revit, o sistema pode ser executado, sem a necessidade de se abrir e ajustar o software Dynamo, utilizando o seu reprodutor nativo. Com isso, a utilização do artefato, demanda necessariamente a utilização do software Revit, mesmo que o modelo esteja em IFC.

4.2 OBJETO DO ESTUDO

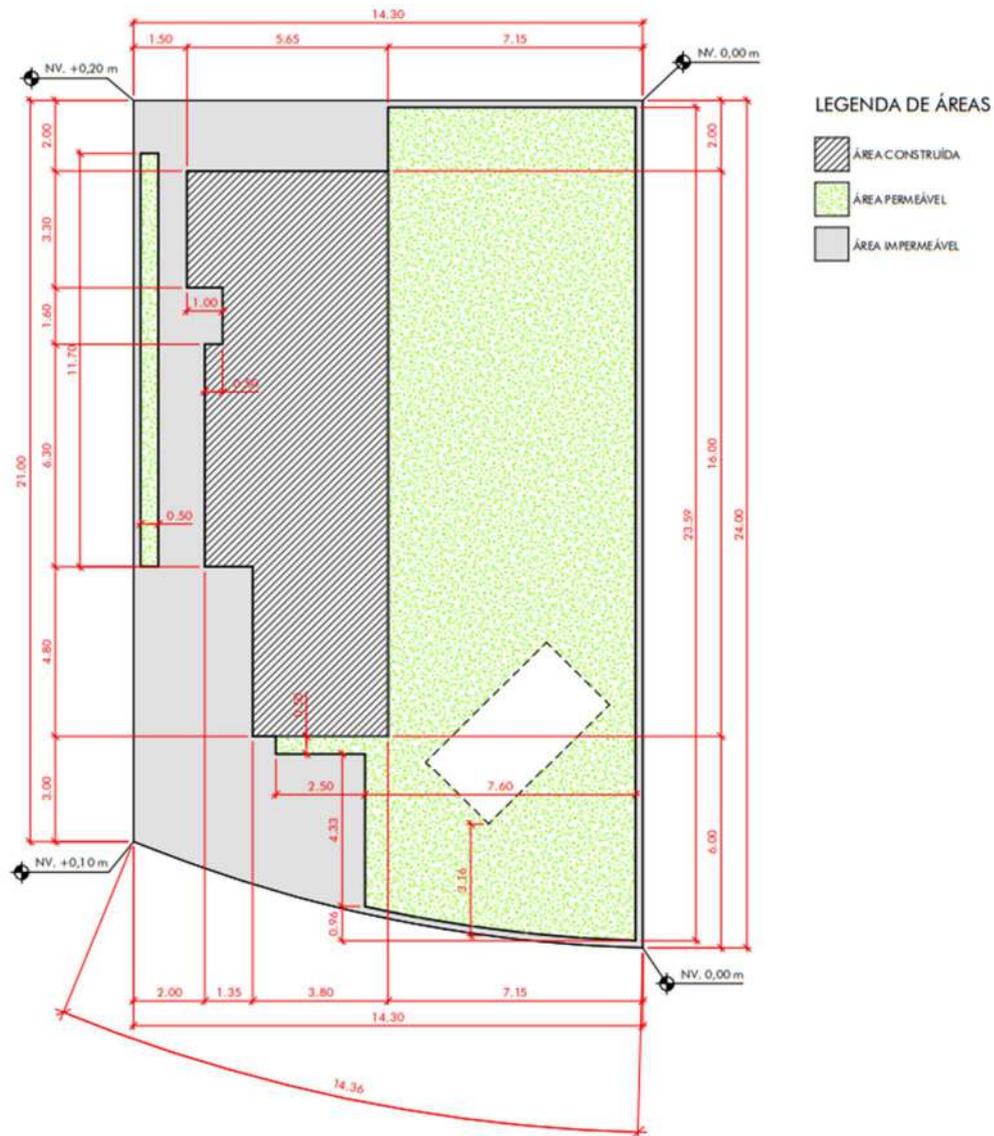
O objeto de estudo é um projeto de construção residencial unifamiliar, aprovado pelo autor na cidade de Poços de Caldas, na data de 16 de novembro de 2020.

Trata-se de uma edificação térrea, de aproximadamente 75 metros quadrados, localizada na Avenida Alana Silvia Santos, Lote 02, Quadra 09 do bairro Jardim Bandeirantes, Figura 23. O macrozoneamento no local é uma Zona de Adensamento Restrito (ZAR), possuindo os seguintes parâmetros urbanísticos no que se refere à legislação vigente:

- Coeficiente de aproveitamento de 1,20;
- Taxa de ocupação de 50%;
- Taxa de permeabilizada máxima de 30%;
- Altura máxima da edificação de 16 metros;
- Largura da testada mínima do lote de 12 metros;
- Recuo frontal de 3 metros;
- Afastamento de fundo e laterais de 1,50 metros, desde que a edificação possua até 6 metros de altura.

O projeto específico foi elaborado com o auxílio do software Autodesk Revit 2019, o qual foi atualizado para a versão 2023, feita de forma automática pelo próprio software. Dentro do projeto, foi criado um piso, representando a totalidade do terreno e toda a área construída, feita com o comando de área, do próprio Revit. Essa área, por sua vez, leva o nome de “Área Construída”.

Figura 23 Implantação



Fonte: Autor

4.3 VERIFICAÇÃO DE MODELO BIM: ESTUDOS

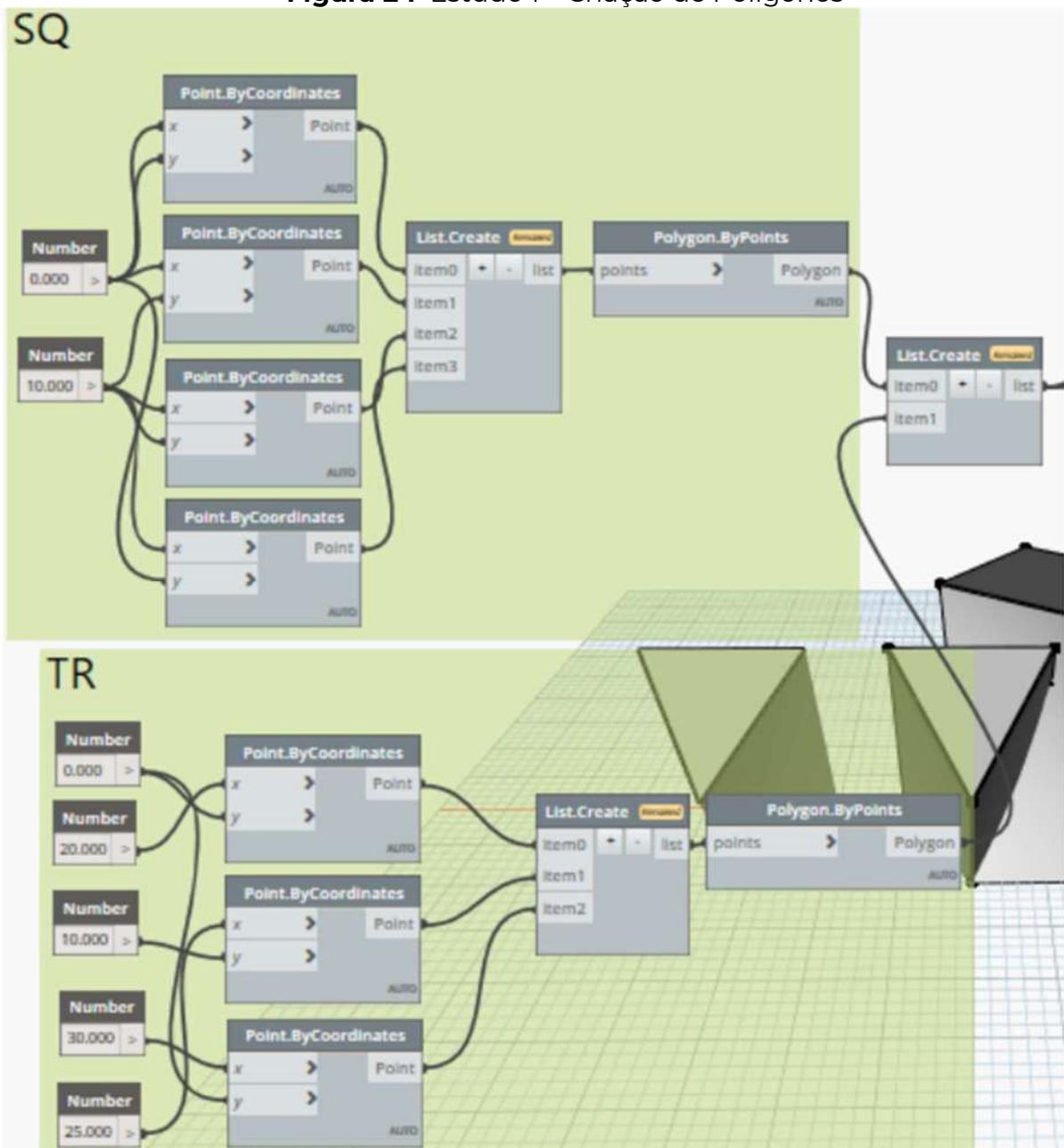
Todos os parâmetros urbanísticos, indicados no Anexo A, podem ser parametrizados, uma vez que são lógicos e dizem respeito a atribuições numéricas. Porém, a título de experimentação e avaliação da ferramenta, são utilizados o recuo frontal e afastamentos laterais e de fundo.

Para tanto, são realizados a seguir, testes para entendimento e absorção da dinâmica entre os programas Dynamo e Revit, com o objetivo de se buscar um sistema otimizado.

- Estudo 1

Inicialmente foram criados quatro pontos com o nó *Point.ByCoordinates* que necessita de duas entradas numéricas para X e Y, criadas com o nó *Number*. Todos esses números não aninhados em uma lista, com o nó *List.Create* e um polígono é feito com base na lista, com o nó *Polygon.ByPoints*, Figura 24.

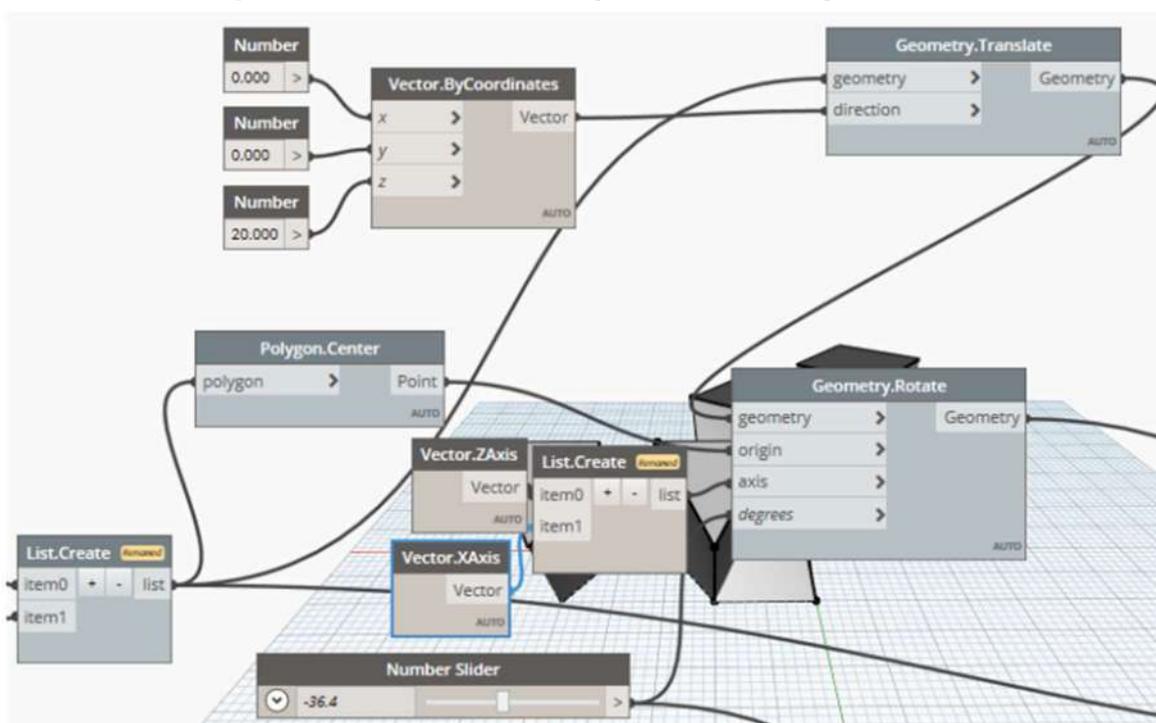
Figura 24 Estudo 1 – Criação de Polígonos



Fonte: Autor

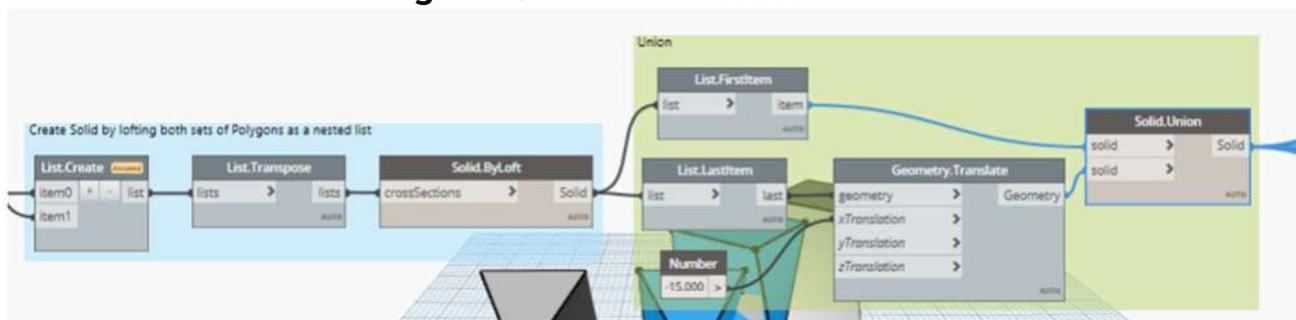
Os mesmos procedimentos são feitos com mais três pontos, formando um triângulo. Tanto o quadrado inicial quanto o triângulo são aninhados em uma lista. Na sequência, cria-se uma superfície com base nesses polígonos e uma cópia delas é criada, utilizando-se o nó *Geometry.Translate*, com base na entrada de um valor vetorial, em qualquer uma das coordenadas X, Y e Z. A superfície também é rotacionada, utilizando-se o nó *Geometry.Rotate*, Figura 25.

Figura 25 Estudo 1 – Rotação e Transladação de Geometria



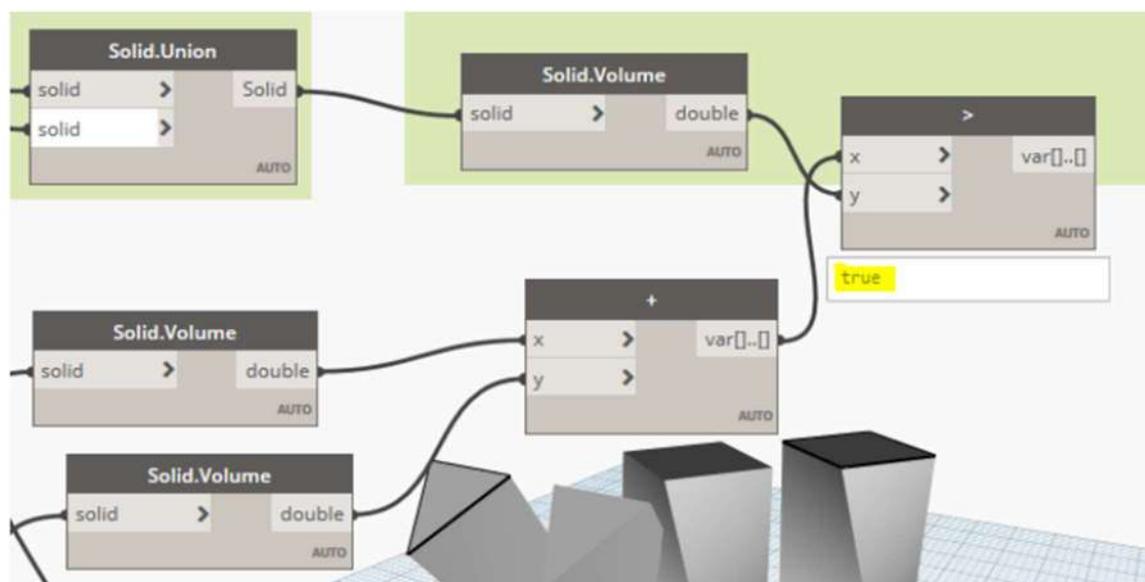
Fonte: Autor

Com base nessas duas superfícies criadas, foram gerados sólidos utilizando-se o nó *Solid.ByLoft*, unindo o polígono quadrangular com o triangular, através do nó *Solid.Union*. Para verificar se havia intersecção entre os sólidos foram criados prismas de forma separada, Figura 26.

Figura 26 Estudo 1 – União dos sólidos

Fonte: Autor

Em seguida, foram analisadas as intersecções entre os sólidos, com base no volume dos elementos. Caso o volume dos objetos interseccionados fosse menor que o volume dos objetos originais, através de um verificador matemático, uma saída de *True* ou *False* seria apresentado, sendo *True* para caso o programa identificasse a intersecção e *False* para caso não houvesse, Figura 27.

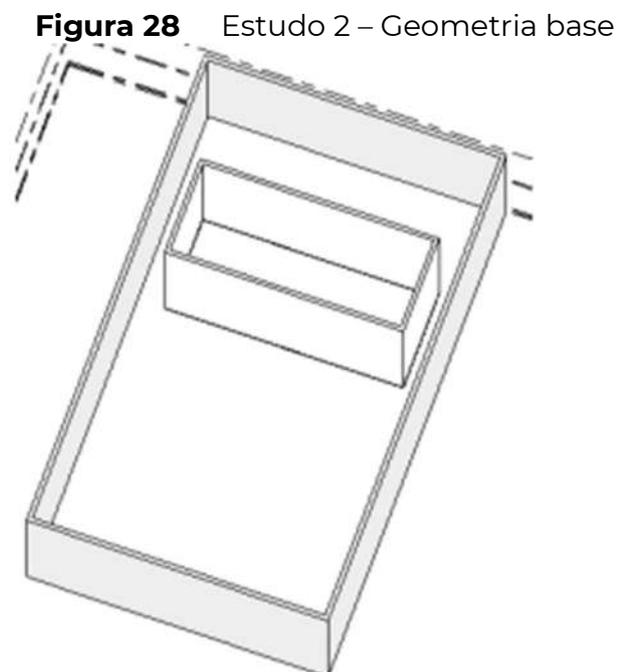
Figura 27 Estudo 1 – Verificação de Intersecção

Fonte: Autor

- Estudo 2

Com os conhecimentos obtidos no estudo anterior, os testes envolvendo o Revit e modelos simples é iniciado. Para isso, é criado um projeto contendo geometrias simples, com quatro paredes externas,

representando os muros de divisa de uma edificação e quatro paredes internas, representando uma edificação, Figura 28.



Fonte: Autor

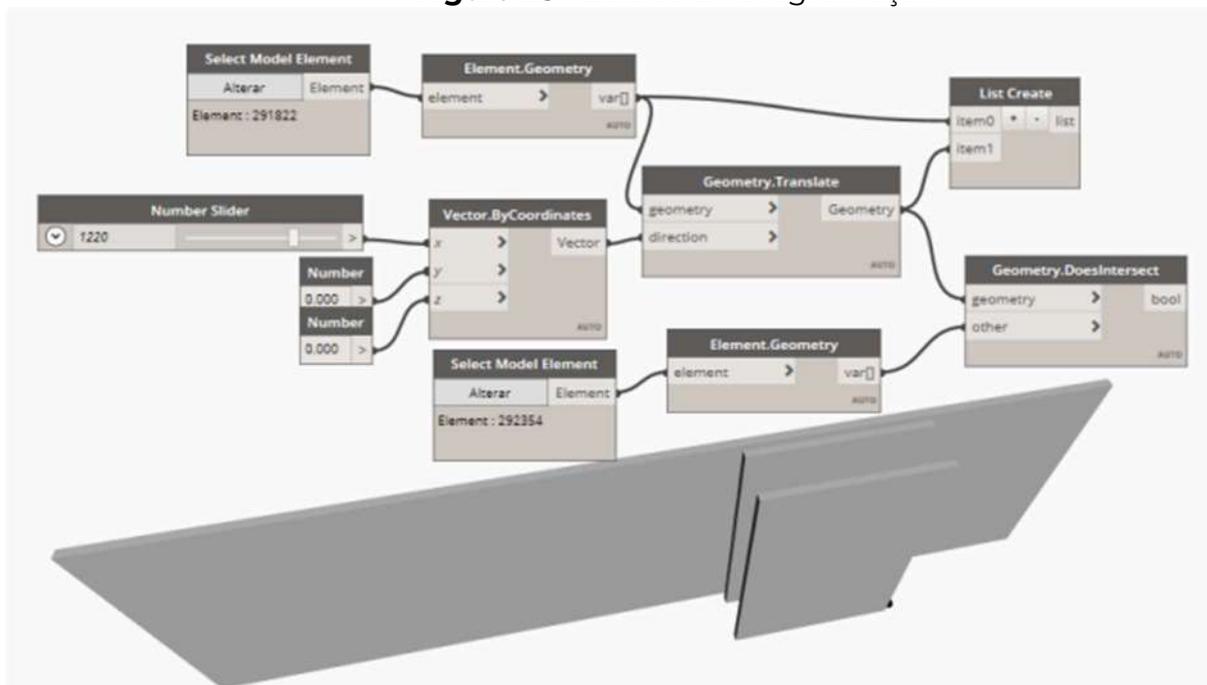
De forma inicial, foi feito o teste de coleta de dados de elementos do Revit, dentro do ambiente da Dynamo. Para a criação de superfície, foi utilizado o nó *Select.PointonFace*, e os quatro vértices de uma parede de divisa foi selecionada. Contudo, não foi possível criar uma superfície com base nos pontos, pois o Dynamo compreende os pontos da base da parede com o topo como sendo o mesmo, por possuírem uma coordenada em comum, além de o mesmo número de séries, por terem sido criadas a partir do mesmo objeto.

Como tentativa de contornar essa situação, foi feito o teste com o nó *Select.Model.Element*, selecionando a parede dentro da interface do Revit. Com o nó *Element.Geometry* é gerada a geometria do Revit dentro da interface do Dynamo, do elemento selecionado no nó anterior.

Com o nó *Geometry.Translate* foi possível deslocar a parede, com a distância desse deslocamento sendo definida através de uma entrada numérica, podendo ser representada pelo valor do afastamento estabelecido em norma. Foi feito nesse estudo também, a verificação de

intersecção com o nó *Geometry.DoesIntersect* que gera o mesmo valor *booleano* indicado no estudo 1, tipo *True* ou *False*, Figura 29.

Figura 29 Estudo 2 - Diagramação



Fonte: Autor

Como essa transposição da geometria é feita com base no eixo da parede, o valor numérico indicado pelo afastamento torna-se impreciso para os fins requeridos, sendo necessária a busca por outra abordagem.

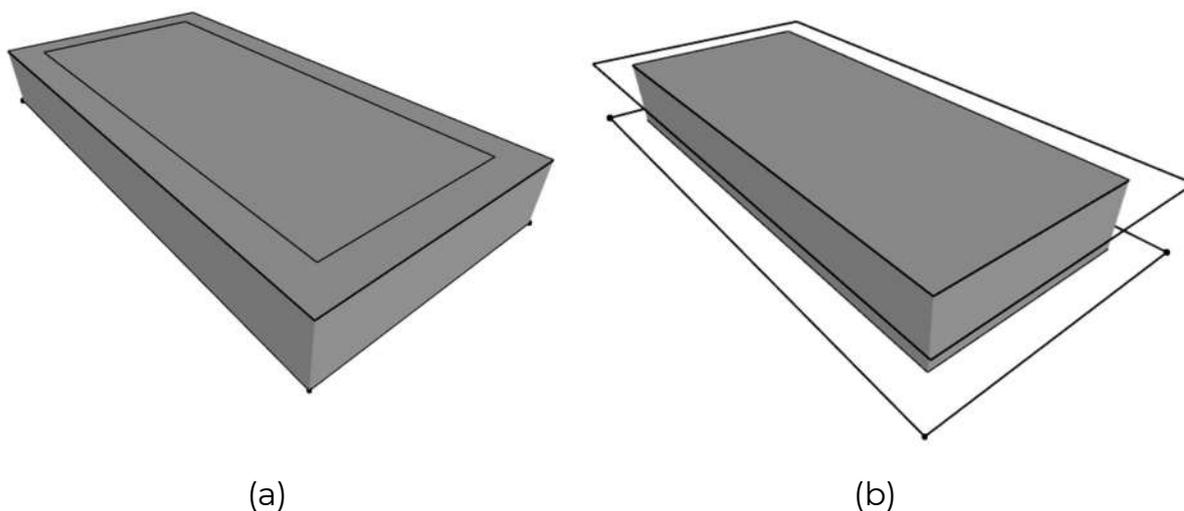
- Estudo 3

Para esse estudo, dentro do Revit foi adicionado ao modelo do Estudo 2 um piso, contornando o terreno. Com o nó *Select Model Element*, seleciona-se o piso e com *Floor.Points*, foi possível obter todos os seus pontos de vértices. Em seguida, com o nó *Polygon.ByPoints* é feita a criação de uma poligonal, passando pelos pontos que compõem o piso. Posteriormente é feito um deslocamento desse polígono com o nó *Geometry.Translate* no eixo Z, com uma entrada numérica. Ambos os polígonos são aninhados em uma lista.

Nesse momento, são criados dois sólidos, com o nó *Solid.ByLoft*. O primeiro é feito com base nos pontos iniciais do piso, Figura 30a. O segundo

foi criado com base num deslocamento interno da geometria principal, criando um sólido menor, com um valor de deslocamento que também pode ser representado pelo afastamento lateral e de fundos de norma, Figura 30b.

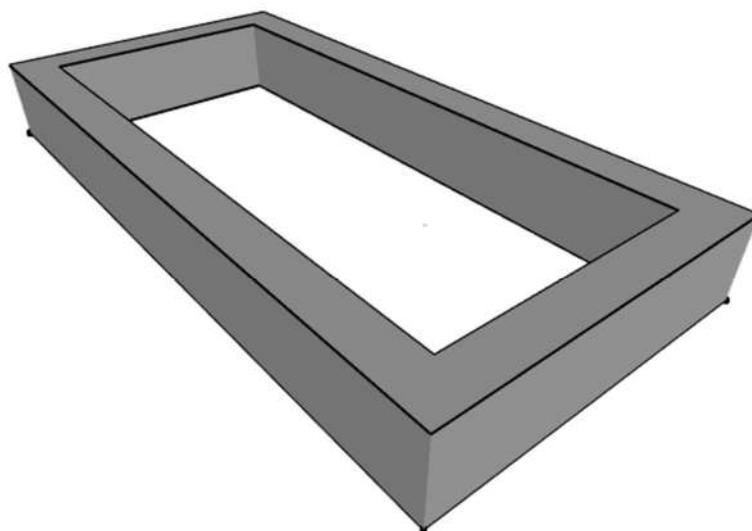
Figura 30 Estudo 3 – Sólidos base criados



Fonte: Autor

Com esses sólidos criados, é feito em seguida a diferença entre eles, com o nó *Solid.Difference* e então uma região representativa dos afastamentos laterais e de fundo é criada, Figura 31.

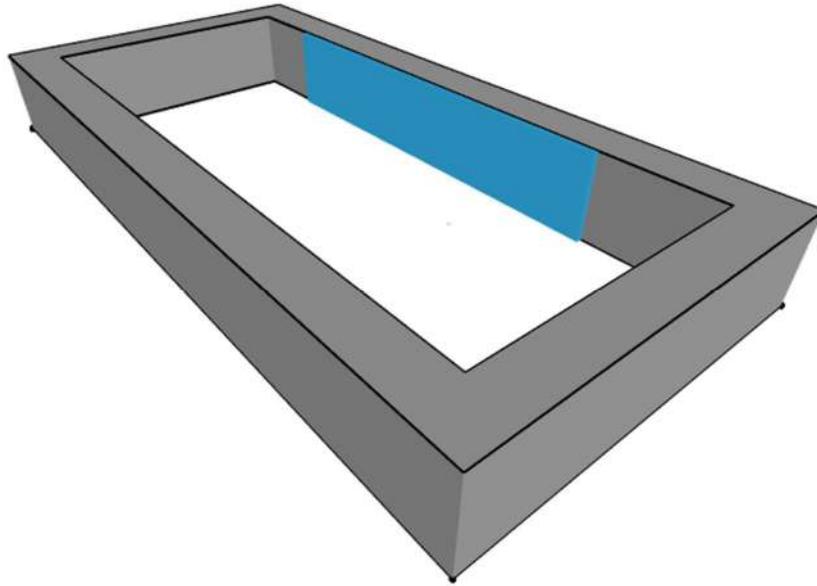
Figura 31 Estudo 3 – Diferença entre os sólidos



Fonte: Autor

Em seguida, foi selecionado uma parede do modelo Revit que representa parte da edificação, com o mesmo comando utilizado para selecionar o piso, isto é, *Select Model Element*. É feita a verificação de intersecção entre as geometrias, com o nó utilizado no Estudo 2, *Geometry.DoesIntersect* gerando um valor tipo *True* quando há intersecção e *False* quando não há, Figura 32.

Figura 32 Estudo 3 – Geometria importada



Fonte: Autor

Com base nos estudos feitos, foi possível verificar a dinâmica entre os softwares Dynamo e Revit e com isso, encontrar uma estratégia adequada para obtenção de um sistema que possua um grau de assertividade aceitável, para o proposto.

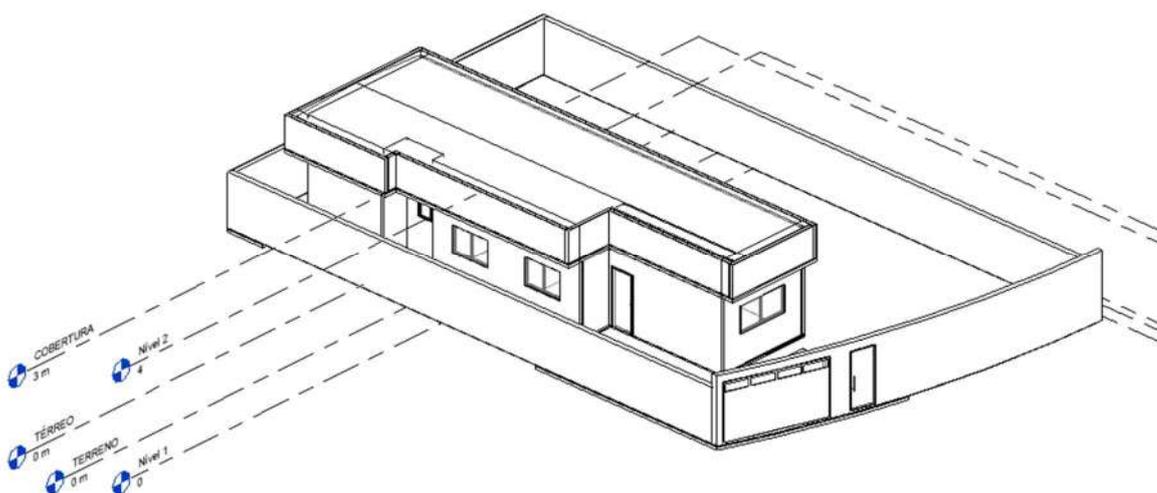
5 DESENVOLVIMENTO DO ARTEFATO

Com base nos estudos realizados nas etapas anteriores, é iniciado então o desenvolvimento do artefato, contemplando todo o conteúdo absorvido nos estudos, acrescidos de recursos provenientes de pacotes extras do próprio Dynamo. Inicialmente, a título de validação dos mecanismos internos do software, é realizado um pré-teste, com o objeto de estudo, indicado no item 4.2.

5.1 PRÉ-TESTE

A princípio, o modelo do objeto de estudo foi aberto dentro do Revit e exportado em formato .IFC 2x3 Coordination View 2.0. Então, um outro projeto foi aberto com base no *template* “Modelo de Arquitetura”, nativo do Revit 2023. O arquivo IFC previamente exportado, referente ao objeto de estudo, foi então importado para esse novo projeto, Figura 33.

Figura 33 Modelo IFC importado



Fonte: Autor

Na sequência, já dentro do Dynamo, são feitas tentativas para a aquisição do modelo .IFC previamente importado dentro do template do Revit. O primeiro teste é feito com o recurso nativo do Dynamo

FileSystem.GetDirectoryContents, com buscas restritas pela *string* “*.ifc”. Para uso correto desse nó, é importante que seja definido pelo usuário uma pasta, em que o arquivo .IFC deve estar, Figura 34. A tentativa não surtiu efeito na importação do modelo para o Dynamo, retornando a saída como *null*, mesmo que na pasta selecionada, encontrava-se apenas o arquivo IFC do objeto de estudo.



Fonte: Autor

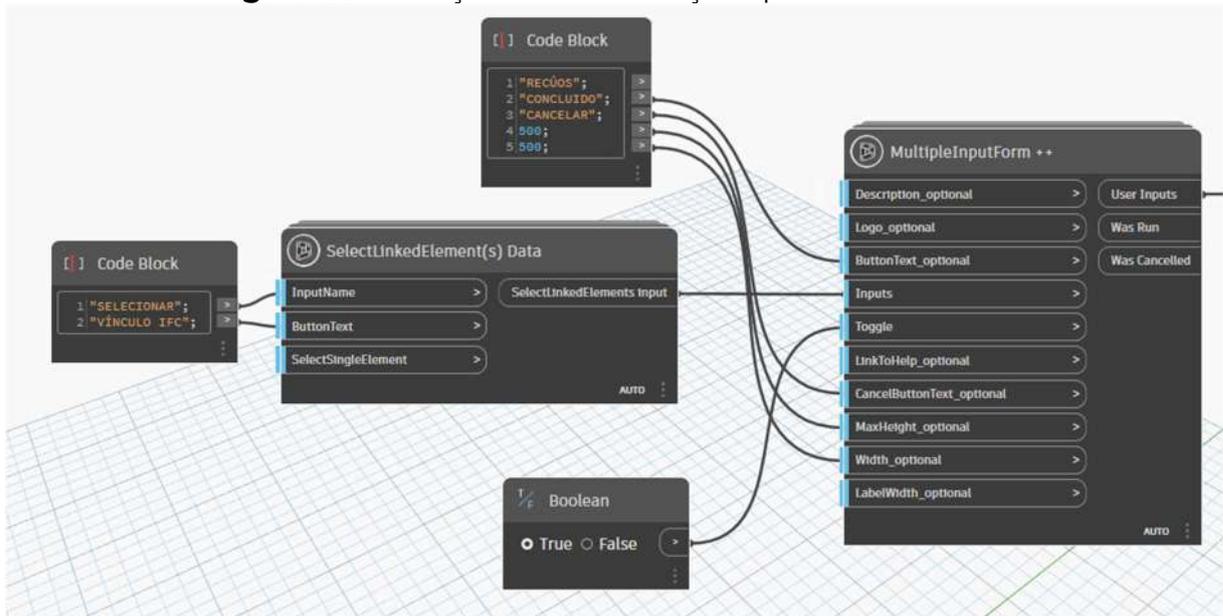
Uma tentativa de seleção de elementos específicos de modelo linkados, foi feita com o uso do pacote “Data-Shapes”, em sua versão 2022.2.103. O pacote, possui a opção de selecionar elementos específicos de objetos linkados (*SelectLinkedElement(s) Data*), além de permitir a criação de janelas iniciais ao usuário, para que sejam feitas as devidas seleções dos valores de entrada ou *inputs* (*MultipleInputForm ++*), Figura 35.



Fonte: Autor

O nó *SelectLinkedElement(s) Data* funciona como um *input* para o nó *MultipleInputForm ++*. Para melhorar a interface do usuário (IU), são feitas as devidas inserções nas entradas de *ButtonText_optional*, *CancelButtonText_optional*, *MaxHeight_optional* e *Width_optional*. Para o nó *SelectLinkedElement(s) Data* as entradas de *InputName* e *ButtonText*, Figura 36.

Figura 36 Inserções de informações para melhoria de IU

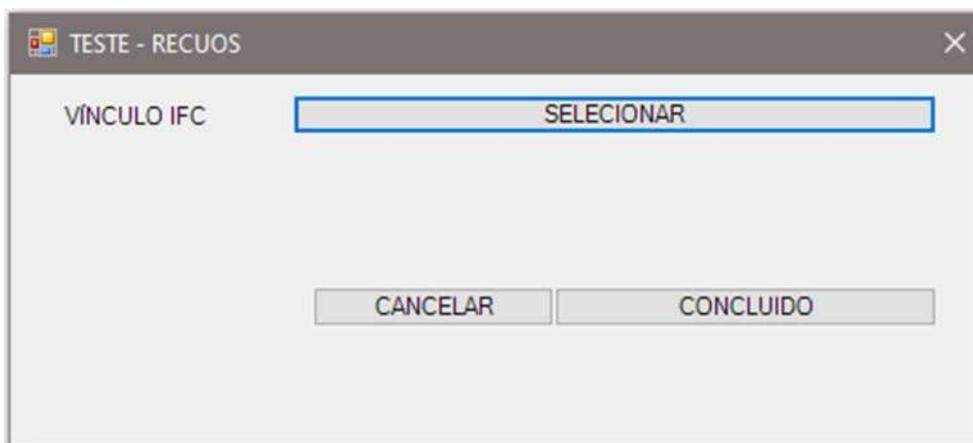


Fonte: Autor

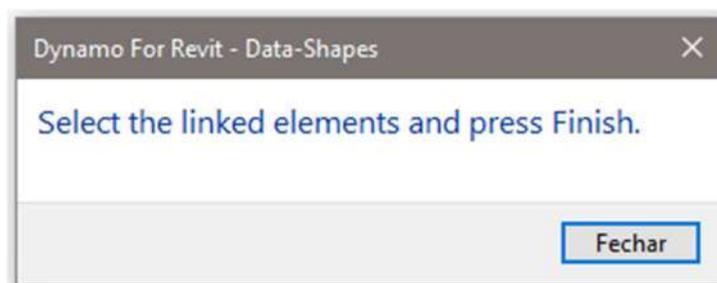
O nó *Boolean* informa ao *MultipleInputForm ++* se o programa irá rodar ou não, através do botão de *True* ou *False*. Os demais nós, trazem as informações textuais que comporão a janela de seleção, como os valores de altura e largura da janela e os textos tanto do formulário quanto em seus botões.

Ao executar o programa, a seguinte janela aparece ao usuário, para que seja selecionado o modelo .IFC, dentro do Revit, Figura 37a. Ao selecionar o vínculo, uma outra janela irá aparecer, solicitando a seleção do objeto requerido pertencente ao vínculo. O objeto em questão é o piso modelado anteriormente, que representa a área limítrofe do lote, Figura 37b.

Figura 37 Janelas de (a) seleção de vínculo e (b) seleção de elementos do vínculo



(a)



(b)

Fonte: Autor

Para melhorar a IU, são feitos ajustes na programação base do nó *MultipleInputForm* ++ em seu nó interno de codificação em Python, Figura 38. Foi substituído o texto “*Select the linked elements and press Finish*” por “*Selecione o Piso e pressione Fechar*”.

Figura 38 Substituição de parte do código fonte do nó

```

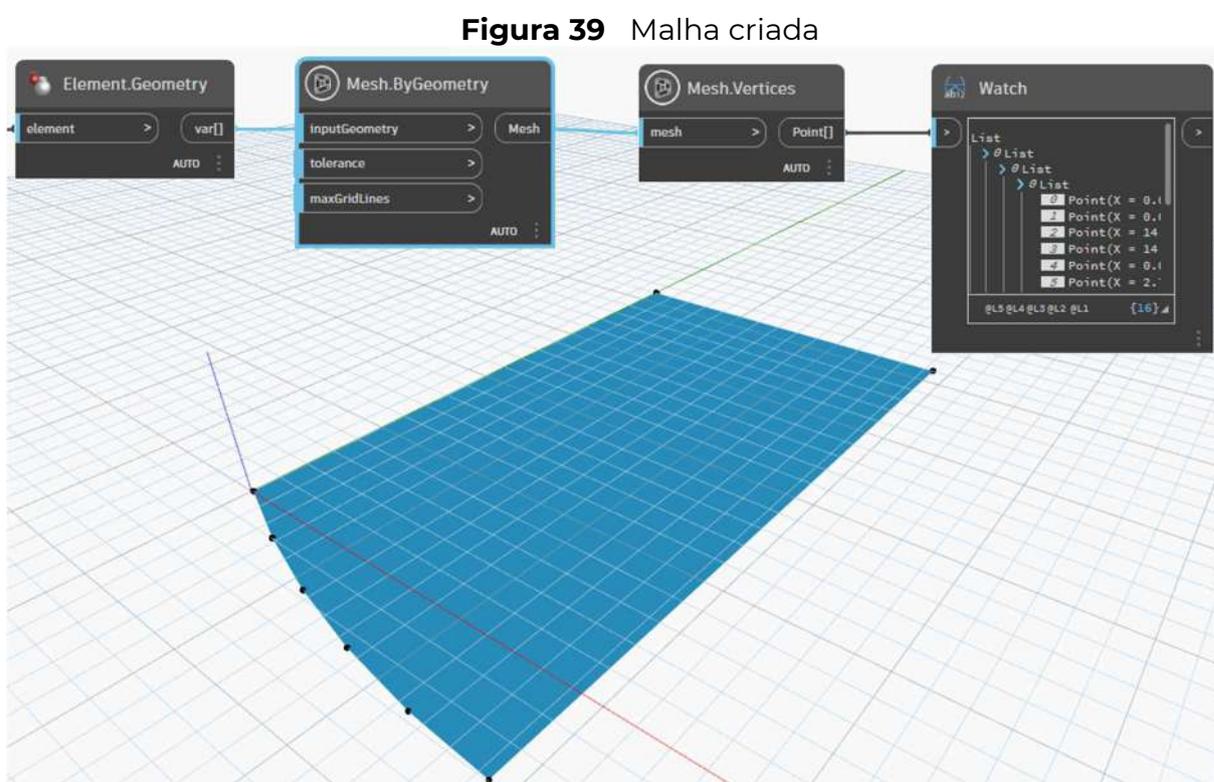
403 def picklinkedobjects(self, sender, event):
404     #This part was made easier by Dimitar Venkov's work
405     for c in self.Controls:
406         c.Enabled = False
407     try:
408         linkref = uidoc.Selection.PickObject(Selection.ObjectType.Element, 'Selecione o Piso e pressione Fechar.')
409         link = doc.GetElement(linkref.ElementId).GetLinkDocument()
410         td = TaskDialog.Show('Seleção de Elemento', 'Selecione o Piso e pressione Fechar.')
411         sel = uidoc.Selection.PickObjects(Selection.ObjectType.LinkedElement, 'Selecione o Piso e pressione Fechar.')
412         selelem = [link.GetElement(s.LinkedElementId) for s in sel]
413         sender.Tag = (selelem)
414     except:
415         pass
416     for c in self.Controls:
417         c.Enabled = True

```

Fonte: Autor

Após a seleção do objeto Piso, que no modelo original é representado por um piso que converge com relação às medidas do lote, é feita a conclusão da seleção, no próprio Revit e depois, a conclusão na janela de *input*.

Com o nó *Element.Geometry*, é possível trazer a geometria importada para o Dynamo e a partir do nó *Mesh.ByGeometry* transformá-la em uma malha, Figura 39. Com o nó *Mesh.Vertices* os pontos de vértice dessa malha são identificados e mostrados na interface do Dynamo. Ambos os nós são provenientes do pacote “MeshToolkit”, versão 3.0.0.

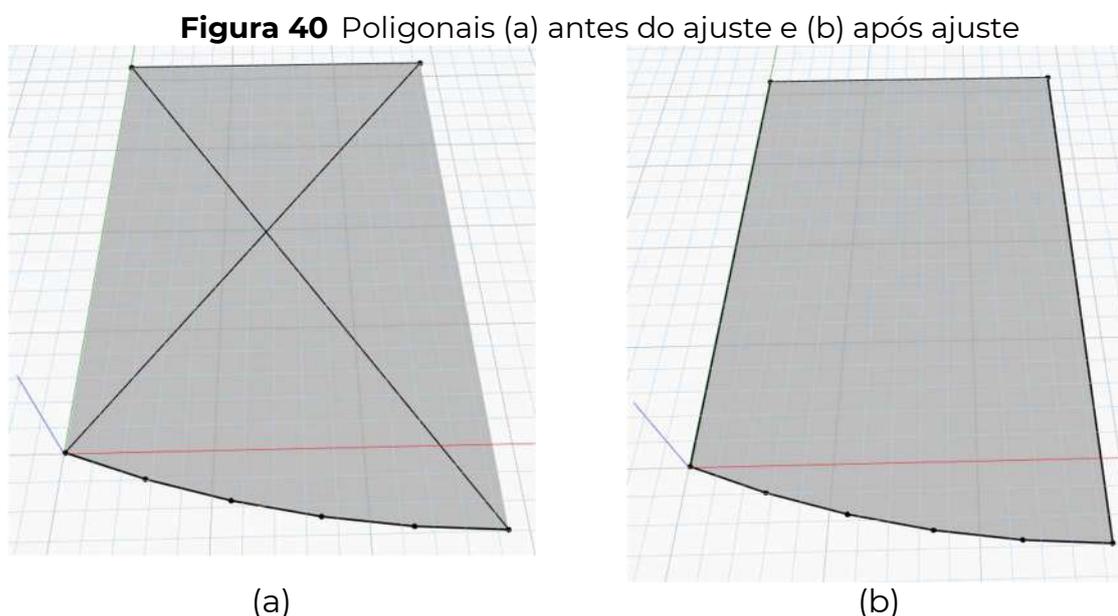


Fonte: Autor

Com o nó *Watch*, é possível identificar o que o nó que o alimenta como entrada está selecionando. A malha em questão aparenta ter oito vértices. No entanto, o nó *Watch* mostra 16 nós, indicando que a ferramenta está trazendo não os vértices da superfície bidimensional, mas sim, os vértices do volume piso, selecionado inicialmente. Para resolver a questão dos nós duplicados e fazer com que o software use somente os nós da superfície, é utilizado o nó *Point.PruneDuplicates*, com uma tolerância de 20

centímetros, suficiente para apagar os vértices que não serão utilizados. Como o Dynamo trabalha com hierarquias de listas, foi necessário planificar os valores, os trazendo para a primeira linha da hierarquia, com o nó *List.Flatten*.

Em seguida, foi criada uma poligonal, com base nos pontos de vértices obtidos anteriormente, com o nó *Polygon.ByPoints*. Como o terreno base possui sua testada curva, o Dynamo estava criando as poligonais a partir de sua identificação, seguindo uma sequência de eixo cartesiano de cima para baixo, da esquerda para direita, formando uma poligonal que não seria passível de ser extrudada e não funcionaria para os objetivos desse estudo, Figura 40a. Um ajuste precisou ser feito com relação às coordenadas dos pontos, para que o Dynamo fosse capaz de os identificar corretamente e criar uma poligonal coerente com o terreno, Figura 40b. O ajuste foi feito com o nó *List.Reorder*, de maneira que o usuário possa utilizar o reconhecimento do limite do terreno de forma automatizada.

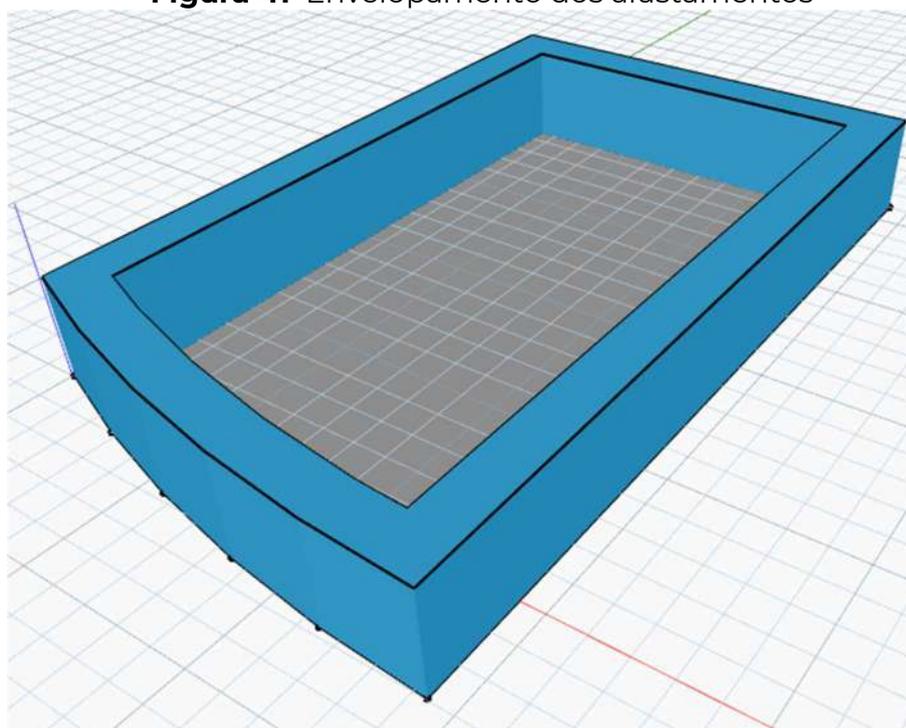


Fonte: Autor

Após o correto desenvolvimento da poligonal, são feitos os mesmos procedimentos elencados no Estudo 3, do item 4.3, em que são criados deslocamentos da poligonal, tanto no sentido vertical (eixo Z), quando deslocamentos internos, no sentido do plano, representados pelos

valores de afastamentos pré-estabelecidos em norma. Também são criados sólidos com o nó *Solid.ByLoft* e é feita a subtração dos sólidos com o nó *Solid.Difference*, chegando no resultado de um envelopamento do afastamento lateral e de fundo, Figura 41.

Figura 41 Envelopamento dos afastamentos



Fonte: Autor

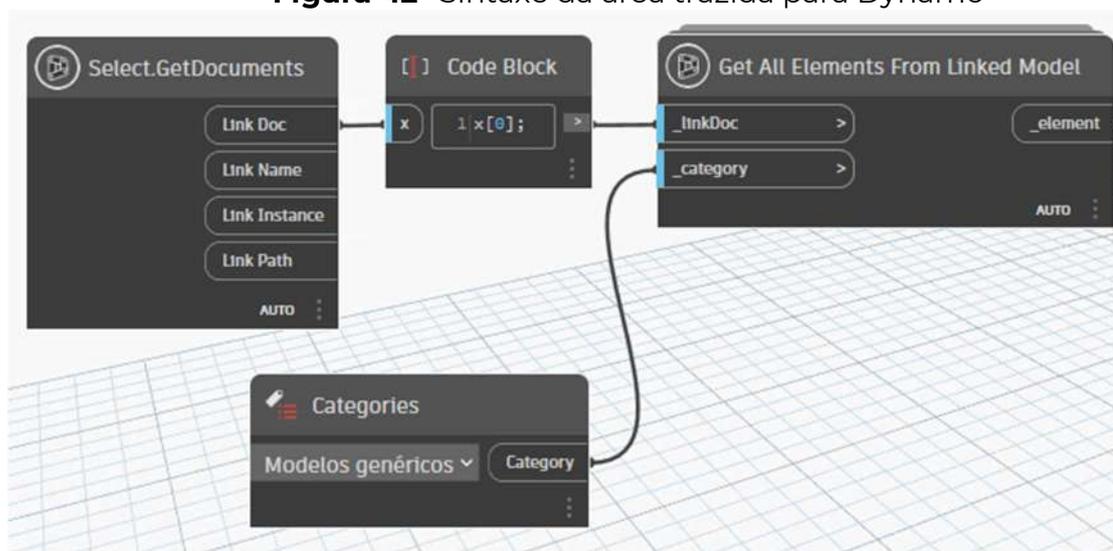
Em seguida, é necessário importar os elementos que serão utilizados para validação dos afastamentos, ao serem conflitados com o envelopamento criado. Para tanto, torna-se necessário a escolha correta do tipo de elemento a ser utilizado, na tentativa de se paginar as diversas possibilidades projetuais.

Os elementos de “Modelos Genéricos” são selecionados, pois dentro do Revit, ao se criar elementos do tipo “Área”, ele é traduzido para o IFC como tal. Os elementos do tipo área possuem um comportamento relevante para a proposta de checagem, uma vez que podem ser descritos em projeto facilmente e irão possuir exatamente a geometria que se espera verificar.

Com o pacote “archilab”, em sua versão 2023.213.1523, é possível fazer seleção de forma automática de elementos com filtros de categorias,

com os nós *Select.GetDocuments* e *Get All Elements From Linked Model*. Com esses comandos e o devido filtro de categoria, selecionando “Modelos Genéricos”, tem-se como resultado diversos itens que não são pertinentes à checagem, como diversas portas e janelas, Figura 42.

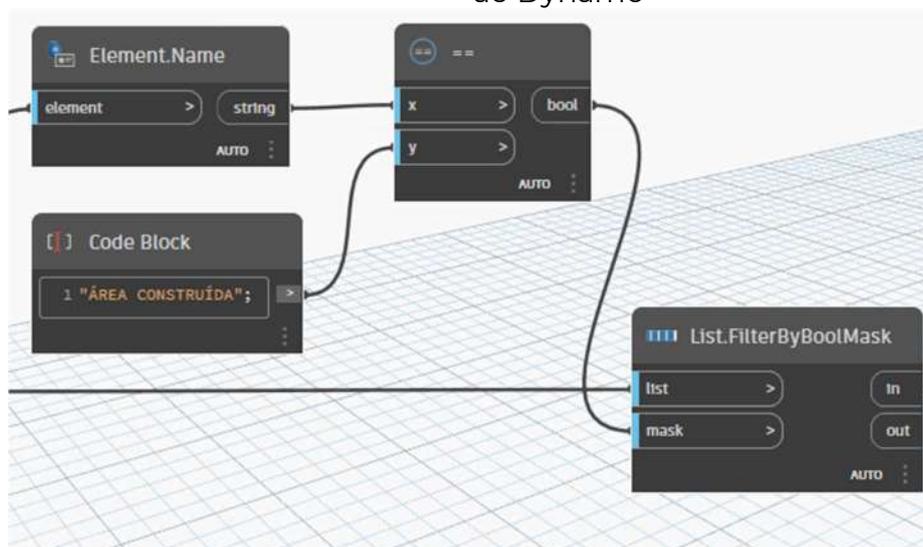
Figura 42 Sintaxe da área trazida para Dynamo



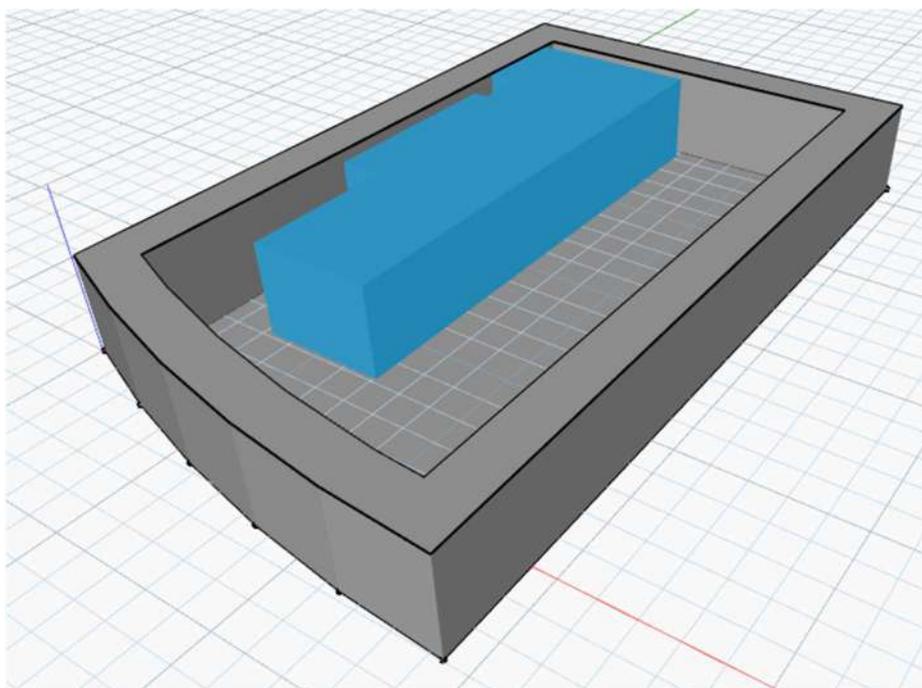
Fonte: Autor

Nessa listagem de elementos, com o nó *Watch* é possível verificar que existe o modelo genérico proveniente da modelagem da área no Revit, chamada “Área Construída”, pré-estabelecido em etapa anterior. Em seguida, é realizado um filtro com o nó *List.FilterByBoolMask*, selecionando a *string* requerida, “Área Construída”, para que somente ela seja selecionada, Figura 43a. Com o nó *Element.Geometry* a geometria da área é trazida para o Dynamo, Figura 43b.

Figura 43 (a) Filtro para seleção de modelo genérico e (b) modelo trazido ao Dynamo



(a)



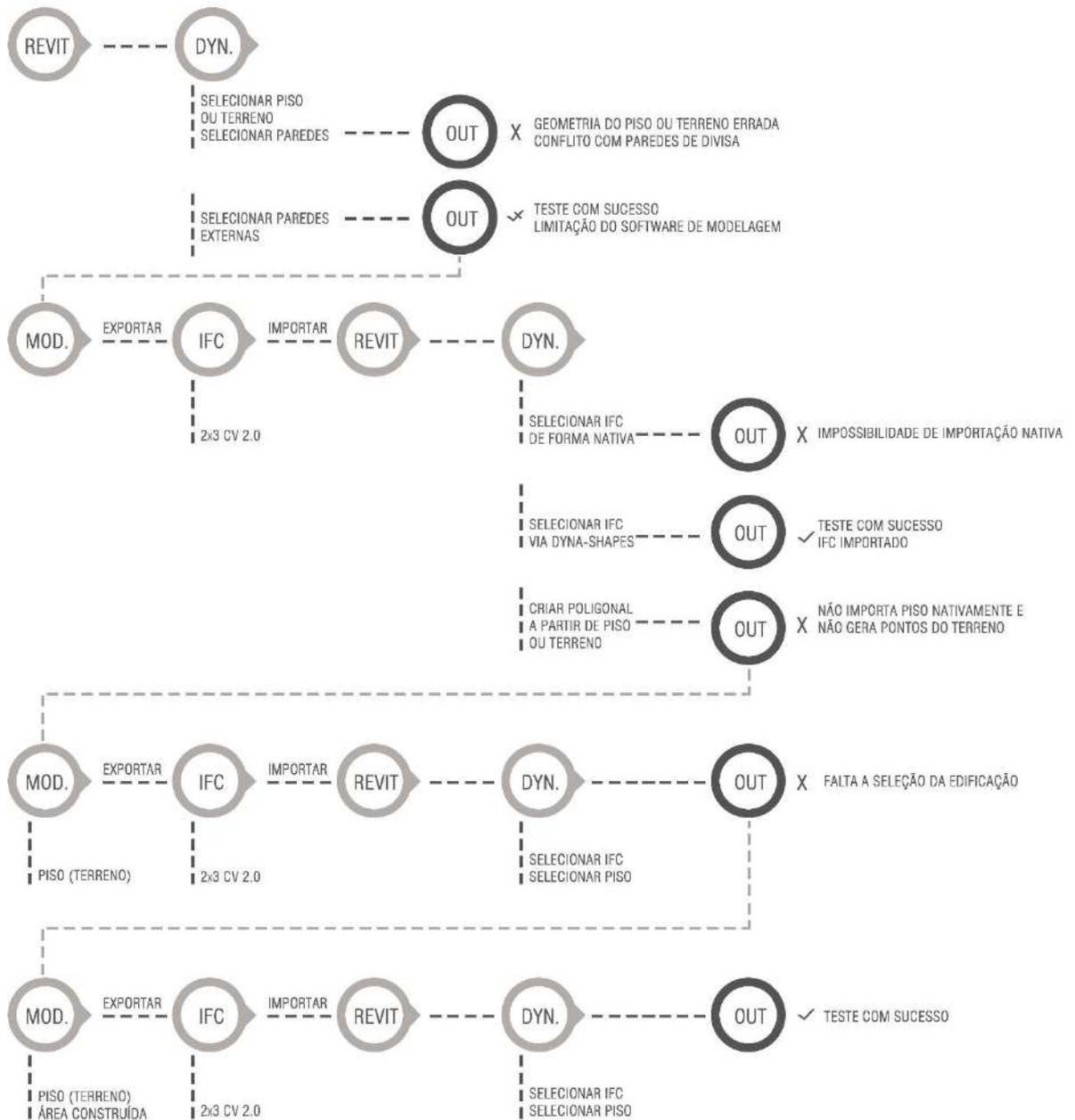
(b)

Fonte: Autor

A verificação de intersecção entre a geometria do envelopamento dos afastamentos e a geometria da área construída é feita com o nó *Geometry.DoesIntersect*. O resultado, a priori, é feito com um nó *Watch*, onde é apresentado o valor de *True* para situações em que os sólidos se encontram e *False* para situações em que os sólidos não se encontram.

O fluxo do pré-teste realizado, está sintetizado na Figura 44 e representa o delineamento recomendado para reprodução e/ou adequação da validação de modelos BIM, utilizando software VPL Dynamo, associado ao software de modelagem BIM, Revit. Pode-se observar no fluxograma os procedimentos realizados que obtiveram sucesso e os que falharam, possibilitando a replicabilidade da proposta.

Figura 44 Pré-teste – fluxo do processo de validação de modelo BIM



Fonte: Autor

5.2 IMPLEMENTAÇÃO

Com os estudos da ferramenta e o pré-teste concluídos, foi feito o desenvolvimento do artefato, seguindo as premissas iniciais estabelecidas anteriormente, agora incorporando a checagem de outros parâmetros urbanísticos.

O artefato final faz as seguintes verificações:

- Afastamentos laterais e de fundo;
- Recuo frontal;
- Altura máxima permitida;
- Área construída;
- Área permeável;
- Taxa de ocupação;
- Coeficiente de aproveitamento;
- Taxa de permeabilidade.

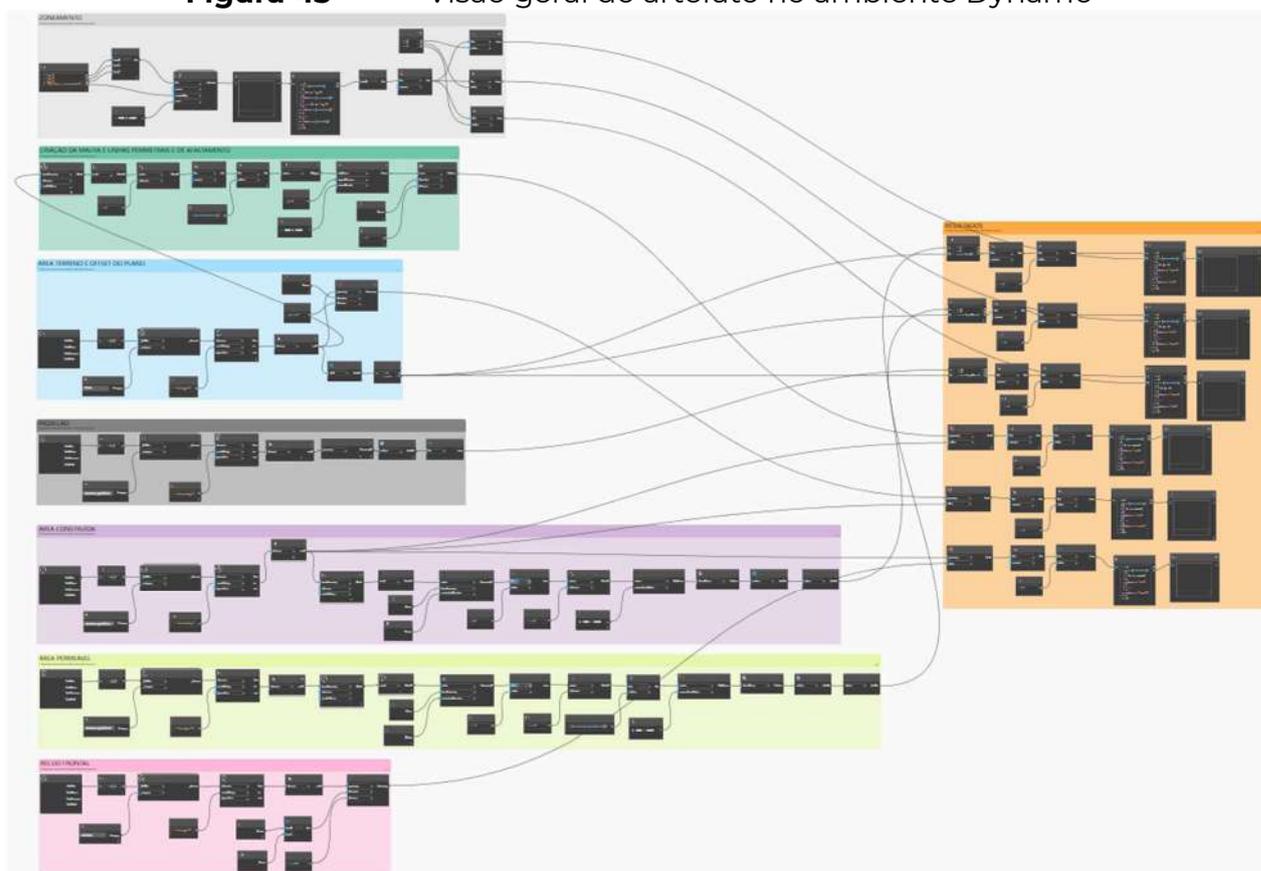
Como tais parâmetros acabam por possuir uma dependência do zoneamento onde o projeto está inserido, torna-se importante a sua seleção, para obtenção dos corretos índices de verificação.

A sistemática adotada no pré-teste, tinha como preceito a importação de forma automática do modelo construído e a seleção manual de um piso, modelado pelo projetista em etapa anterior à checagem, que representaria o terreno. Por se tratar de uma sistemática em que há a necessidade de seleção de um objeto do modelo pelo usuário e conseqüentemente, algo que pode levar a seleções equivocadas, é feita a tentativa de obtenção automática de todas as informações necessárias do modelo.

A entidade área (Área Construída), no pré-teste, é importada ao ambiente Dynamo através de um recurso disponível no pacote “archilad”. Seguindo o mesmo conceito dessa importação, são realizadas alterações no sistema do pré-teste, a fim de se trazer para o Dynamo, especialmente o piso, mas também, outras informações do modelo base.

Com os devidos ajustes em nós específicos, foi possível a obtenção de um modelo mais otimizado e automático. Dentro do ambiente Dynamo, para cada parâmetro urbanístico checado, foram criados grupos com cores distintas para facilitar a visualização da sistemática adotada, Figura 45.

Figura 45 Visão geral do artefato no ambiente Dynamo



Legenda:



Fonte: Autor

5.2.1 Zoneamento

Os parâmetros urbanísticos variam de acordo com o zoneamento, como área permeável, área construída, altura máxima da edificação, entre outros. Portanto, torna-se importante que no artefato proposto, haja a

validação do zoneamento para correta seleção dos valores dos parâmetros a serem verificados.

O primeiro teste para importação do zoneamento do projeto para o ambiente Dynamo, foi pensando de forma automática, com o usuário, inserindo em campo específico do modelo a nomenclatura referente à zona em questão. Seguindo a sistemática proposta para a implantação do artefato, foi proposto que houvesse a inserção no campo “Dados de Identidade” no Revit, da sigla do zoneamento. Foram testadas diversas formas de exportação do modelo base arquitetônico, com distintos MVD’s. Contudo, nesse primeiro teste, tal informação não foi possível de ser identificada pelo Dynamo.

Como o primeiro teste de obtenção do zoneamento de forma automática pelo artefato não obteve êxito, partiu-se para outra estratégia. Dessa vez, sem a necessidade de qualquer inserção no modelo, mas sim com a seleção do zoneamento de forma manual, pelo projetista, no momento da execução do artefato de checagem.

Após buscas pela comunidade do Dynamo, foi descarregado dentro do artefato, no ambiente Dynamo, o pacote *SpingsUI* em sua versão 132.1.1 que possibilita o uso de listas suspensas, através do nó *List.DropDown*. Com um nó *code block* foram definidos a título de validação do recurso proposto, três zoneamentos, cada um com os seus devidos parâmetros inseridos. Em seguida, foram realizados os primeiros testes, em que ao executar o artefato, uma janela aparece ao usuário, com uma lista suspensa e a possibilidade de seleção do zoneamento.

Após a validação da funcionalidade da lista suspensa, é necessário que o artefato seja capaz de interpretar a seleção do usuário e na sequência buscar em um banco de dados os corretos valores dos parâmetros urbanísticos. Para isso, criou-se um nó *code block* contendo as informações dos três zoneamentos propostos, sendo que, após a execução do artefato e seleção do zoneamento, os valores são inseridos em uma lista e para cada parâmetro é atribuído um índice, que varre a lista criada, trazendo esses valores para as próximas etapas, Figura 46.

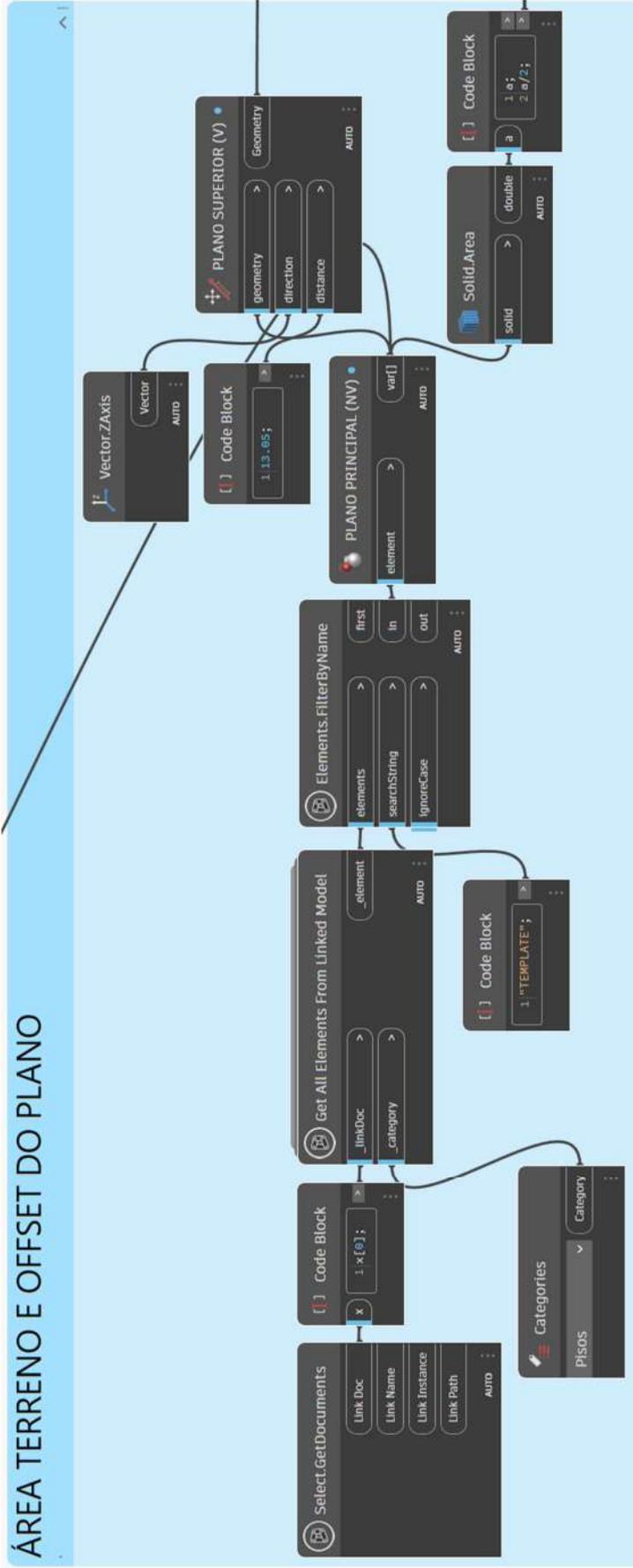
5.2.2 Área do terreno e offset do plano

Para importação do piso, que nesse estudo de caso específico, perfaz a totalidade do terreno da construção e foi previamente modelado no arquivo base, foi seguida a sistemática proposta para implantação. Foi utilizado o nó *Select.GetDocument*, com categoria principal de *Pisos* selecionada, tendo a possibilidade da criação de um filtro. Tal recurso possibilita a seleção de um item específico dentro de uma categoria. O filtro em questão é uma sequência de caracteres, denominada *string*, com o valor de “Template”, inserida via nó *Elements.FilterByName*. Com essa filtragem, o artefato compreende que dentro de uma seleção de categoria, nesse caso pisos, ele irá importar exclusivamente as informações de pisos que contenham a *string* “Template”, em seu nome.

A modelagem, dentro do ambiente Dynamo, da área do terreno, faz uso do nó *Solid.Area*. O valor encontrado para área se mostrou como sendo exatamente o dobro do esperado, inferindo que o nó traz a área de ambas as faces do modelo. Portanto, foi feita a divisão do valor encontrado por dois, para correta validação de todos os parâmetros urbanísticos propostos.

A verificação da altura máxima da edificação, é feita com base no modelo obtido em etapa anterior, agregando-se um deslocamento no eixo “Z” (altura), com o valor dentro do esperado pelo zoneamento. O nó utilizado para tal deslocamento (*offset*) é o *Geometry.Translate*, com o nó *Vector Z Axis* para indicação do vetor de translação, Figura 47.

Figura 47 Grupo Área do terreno e offset do plano



Fonte: Autor

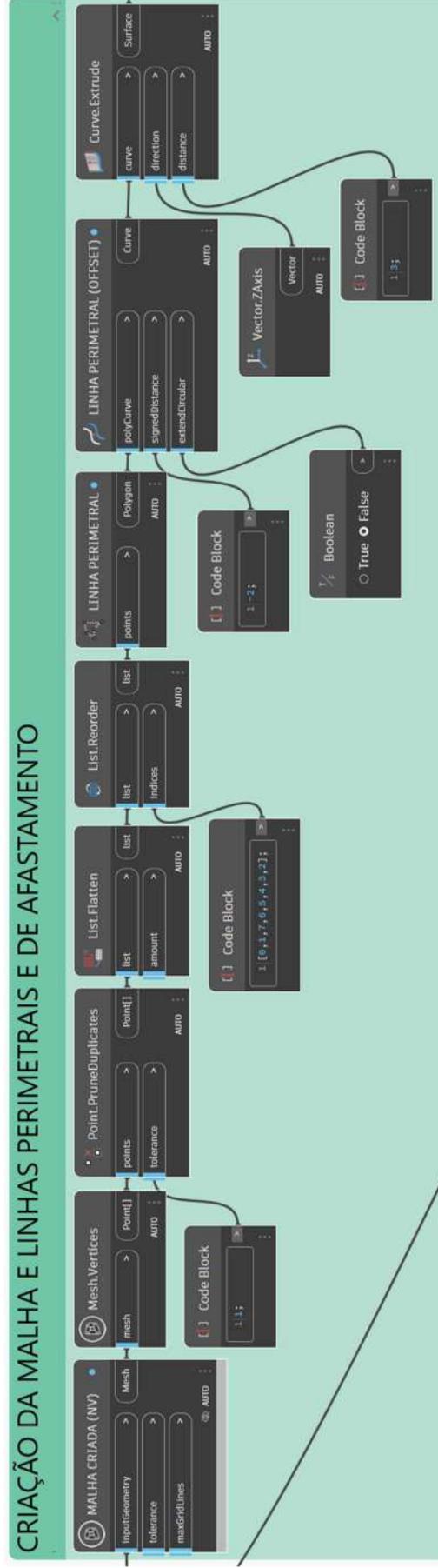
5.2.3 Criação de malha e linhas perimetrais e de afastamento

Para criação das linhas de envelopamento, que serão utilizadas para verificação dos afastamentos, primeiro é necessário a transposição do sólido modelado para criação do terreno em uma malha (*mesh*), com o nó *Mesh.ByGeometry*. Tal alteração se baseia nos achados do pré-teste e tem como intuito, após a criação da malha, na obtenção de uma poligonal que perfaça seu perímetro.

Na sequência, são obtidos os vértices dessa malha, com o nó *Mesh.Vertices* e os pontos de duplicatas apagados com o nó *Point.PruneDuplicates*. Tal exclusão se torna necessária, uma vez que como o sólido modelado possui uma altura e mesmo sendo mínima, em torno de 8 milímetros, pode gerar problemas na obtenção da poligonal. Em seguida, os nós do modelo são rearranjados para criação de uma poligonal fechada, com o nó *Polygon.ByPoints*.

É criada uma cópia com um deslocamento da poligonal criada, com o nó *PolyCurve.Offset*. O deslocamento compreende exatamente no valor imposto pelo valor do recuo lateral e de fundo, específico para o zoneamento. Por fim, é criado um sólido, que compreende no envelopamento dos limites de construção, no que se refere aos recuos laterais e de fundo, com o nó *Curve.Extrude*, Figura 48.

Figura 48 Grupo de malha e linhas perimetrais de afastamentos

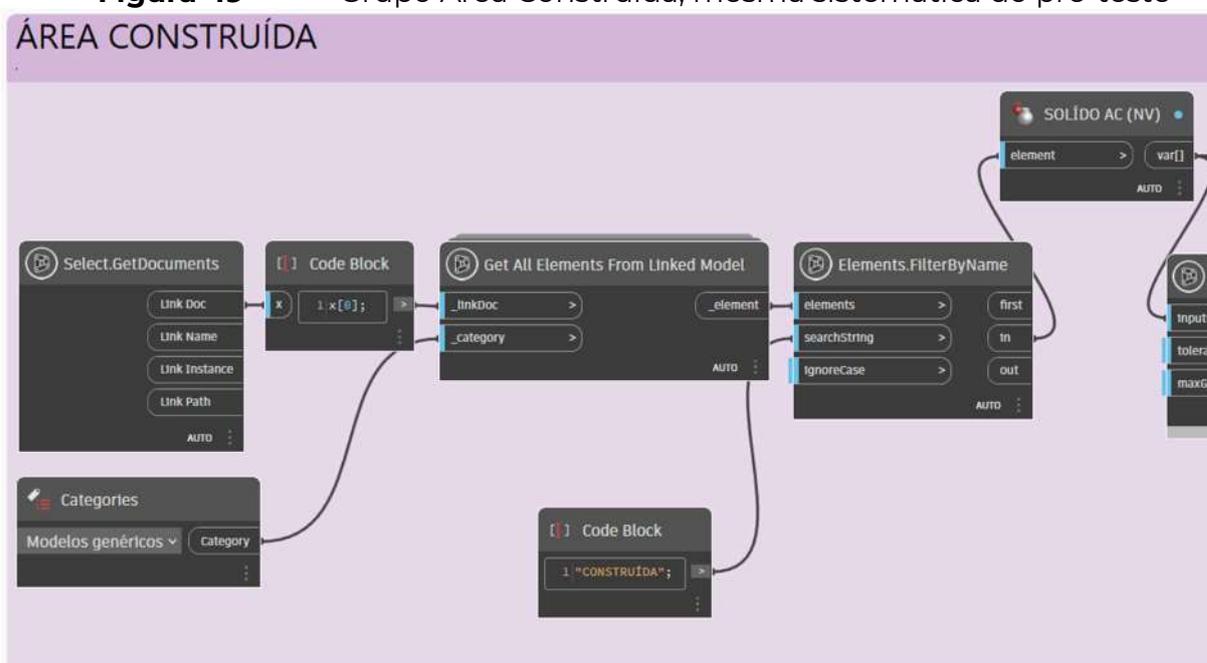


Fonte: Autor

5.2.4 Área construída

Para a importação da área construída do projeto, dentro do ambiente do Dynamo, foram feitos exatamente os mesmos procedimentos elencados no pré-teste, Figura 49. Na sequência, foi necessário a obtenção dos valores referentes à área construída, para checagem dos parâmetros urbanísticos correspondentes, como coeficiente de aproveitamento e taxa de ocupação. Para tanto, foi necessária a manipulação do sólido importado.

Figura 49 Grupo Área Construída, mesma sistemática do pré-teste

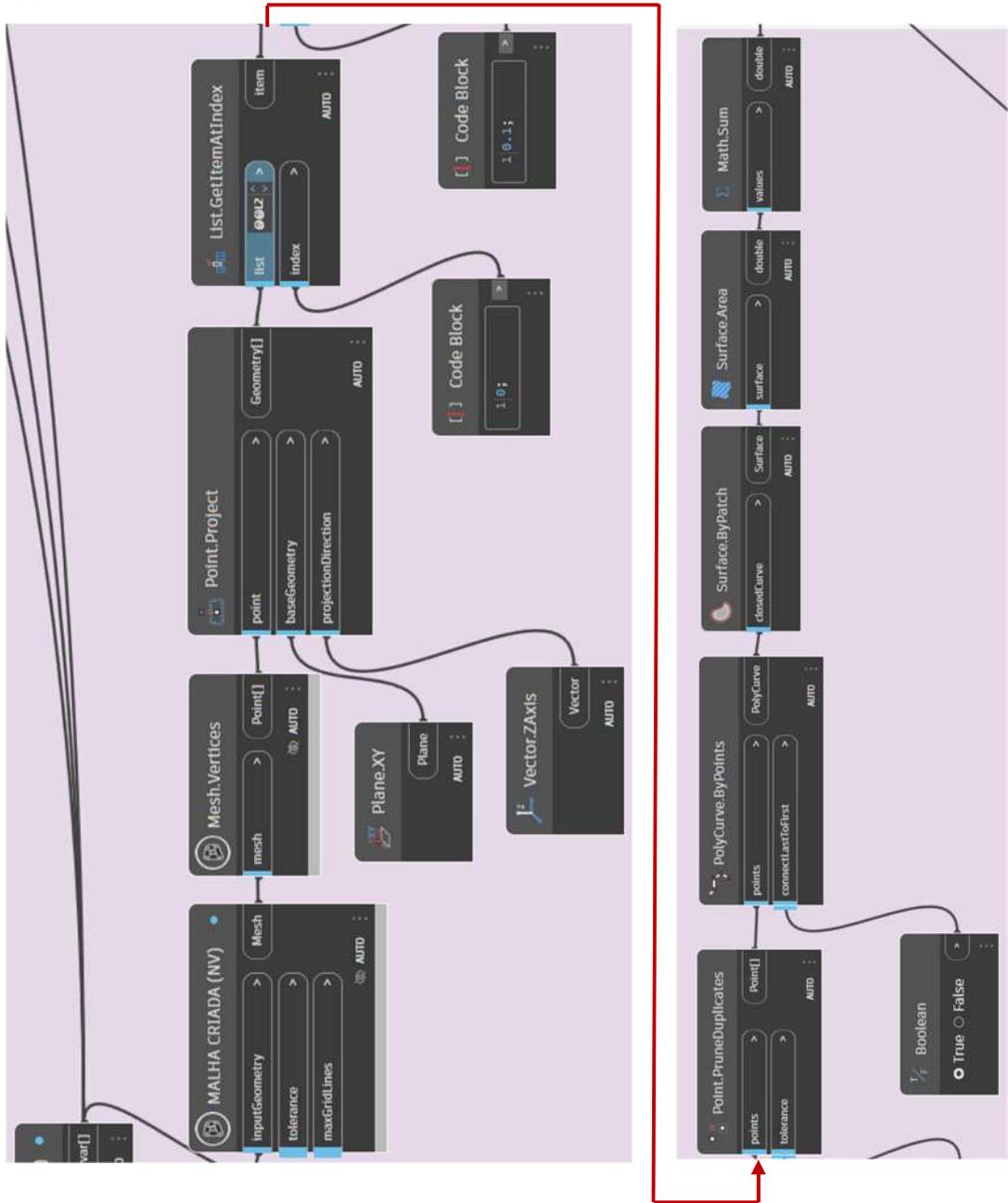


Fonte: Autor

Inicialmente, o sólido foi transposto para uma malha, com todos os seus vértices obtidos, através dos nós *Mesh.ByGeometry* e *Mesh.Vertices*. Em seguida, todos os nós encontrados foram projetados em um plano XY criado, através do nó *Point.Project*, no sentido da altura, isto é, no vetor Z. Com as devidas manipulações de lista, necessárias dentro do ambiente Dynamo, foi criada a poligonal que representa a área construída, com o nó *Polygon.ByPoints*. Por fim, tal poligonal foi transposta em uma superfície, com o nó *Surface.ByPatch*, tendo sua área identificada através do nó *Surface.Area*.

O modelo base arquitetônico possui somente uma projeção de área construída. Contudo, para casos em que existem mais plantas de áreas construídas, torna-se necessário a somatória dos valores, para correta validação do parâmetro de coeficiente de aproveitamento. Para tanto, utilizou-se o nó *Math.Sum*, que faz a soma das áreas das superfícies criadas, Figura 50.

Figura 50 Grupo Área Construída

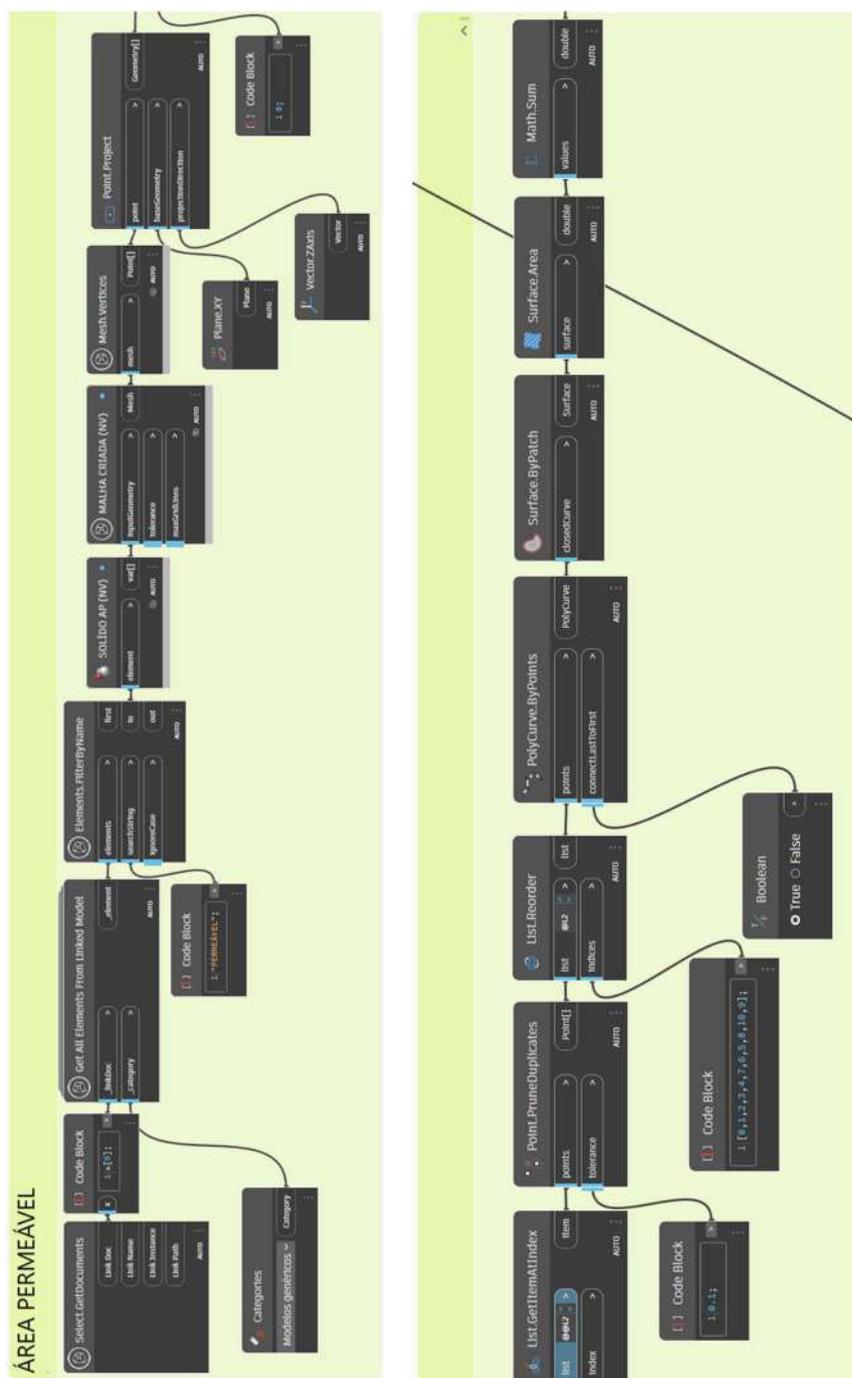


Fonte: Autor

5.2.5 Área permeável e Projeção

A importação da área permeável, para o ambiente do Dynamo, utilizou a mesma sistemática adotada na obtenção da área construída, com a substituição do filtro de valor (*string*) “Construída” para “Permeável” Figura 51.

Figura 51 Grupo de Área Permeável



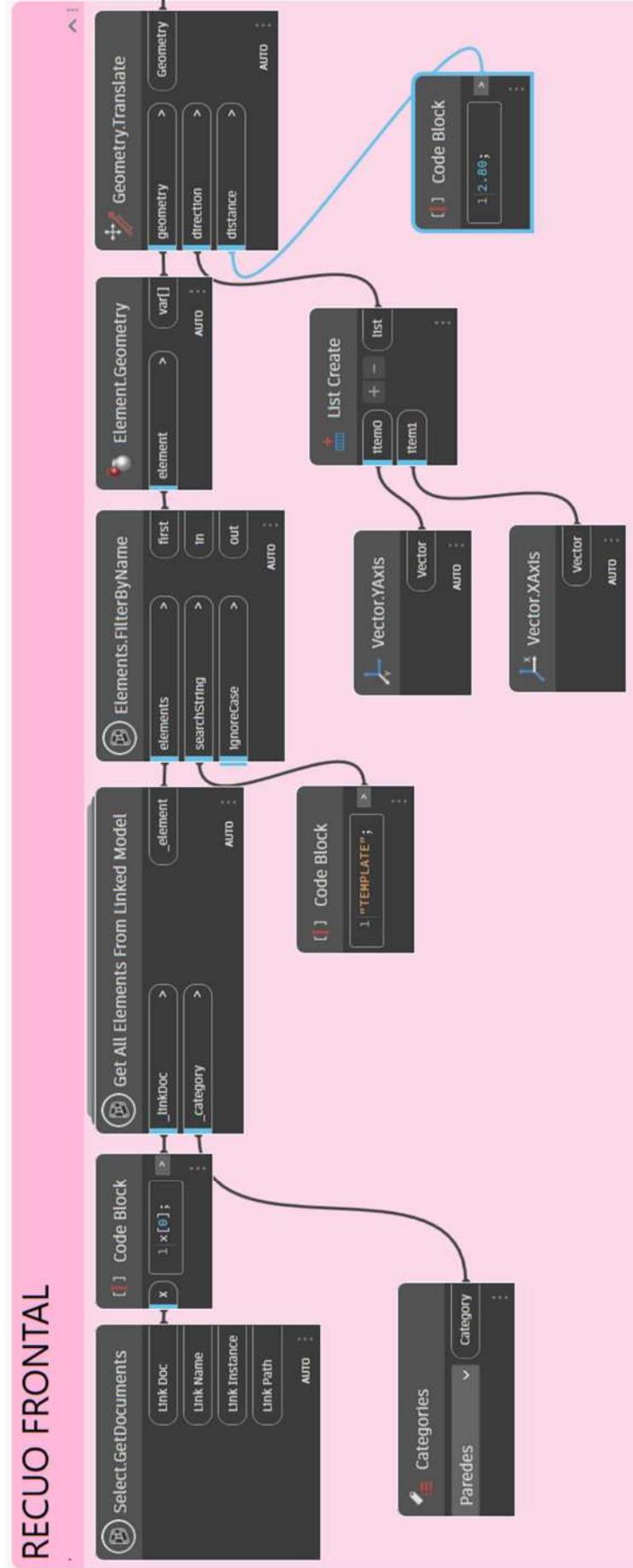
Fonte: Autor

Já o valor que representa a projeção, também é obtida da mesma forma, substituindo o filtro de “Construída” para “Projeção”.

5.2.6 Recuo frontal

A título de resolver a verificação do recuo frontal, foi usado como estratégia a importação do elemento que representa a fachada principal e que está diretamente sujeita à verificação do recuo. Para isso, utilizou-se a mesma seleção adotada para os demais casos, com a categoria parede selecionada e o filtro de *string* “Template”. Feito isso, é gerada a modelagem dessa parede com o nó *Element.Geometry* e é feita sua translação para simulação do recuo frontal, descontando-se o valor da espessura da parede, Figura 52.

Figura 52 Grupo Recuo Frontal



Fonte: Autor

5.2.7 Resultados

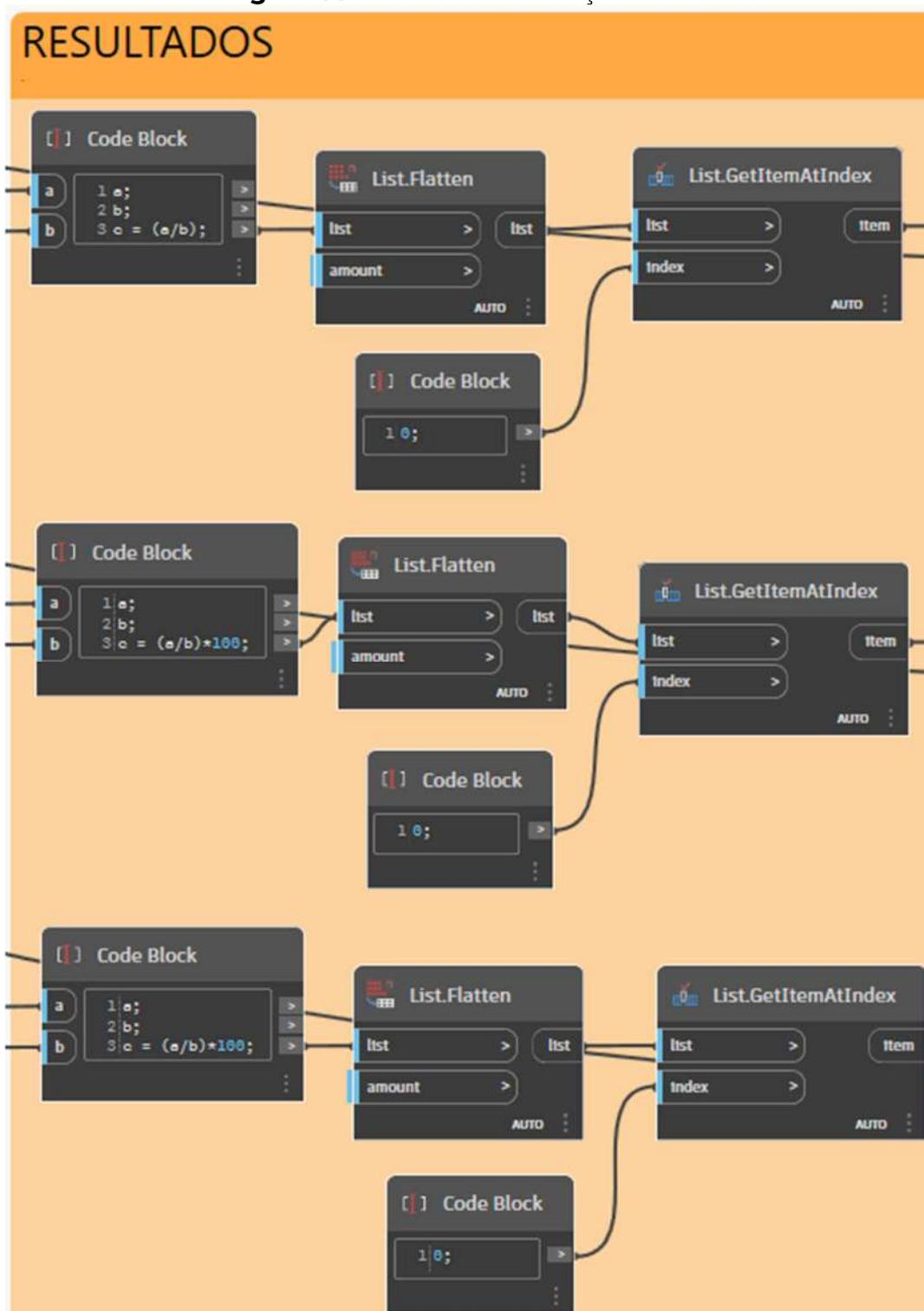
No grupo resultados, é onde encontram-se todas as verificações feitas pelo artefato. Existem no sistema, dois tipos principais de checagens e são elas: (i) checagem numérica dos parâmetros urbanísticos (taxas e coeficientes); e (ii) checagem geométrica de conflitos entre modelos (recuo, afastamentos e altura).

- Verificação numérica

Todas as verificações numéricas conflitam os valores encontrados em projeto, isto é, no modelo base, com os valores pré-estabelecidos durante a seleção do zoneamento, no momento em que se executa o artefato. São verificados, de forma numérica, os seguintes parâmetros urbanísticos: (i) coeficiente de aproveitamento, que é o valor da área construída, dividido pelo valor do terreno; (ii) taxa de ocupação, que é o valor da projeção dividida pela área do terreno, multiplicado por 100 (valor em porcentagem); e (iii) taxa de permeabilidade, que é a área permeável, dividida pela área do terreno, multiplicado por 100 (valor em porcentagem).

Com o auxílio de um nó *code block* são criadas duas entradas “a” e “b”, sendo que a entrada “b” é alimentada pela área do terreno e a entrada “a” é alimentada pelas demais áreas. As devidas equações são feitas conforme o caso, através de uma variável de saída “c”, que na sequência, popula uma lista, Figura 53.

Figura 53 Sintaxe da verificação numérica

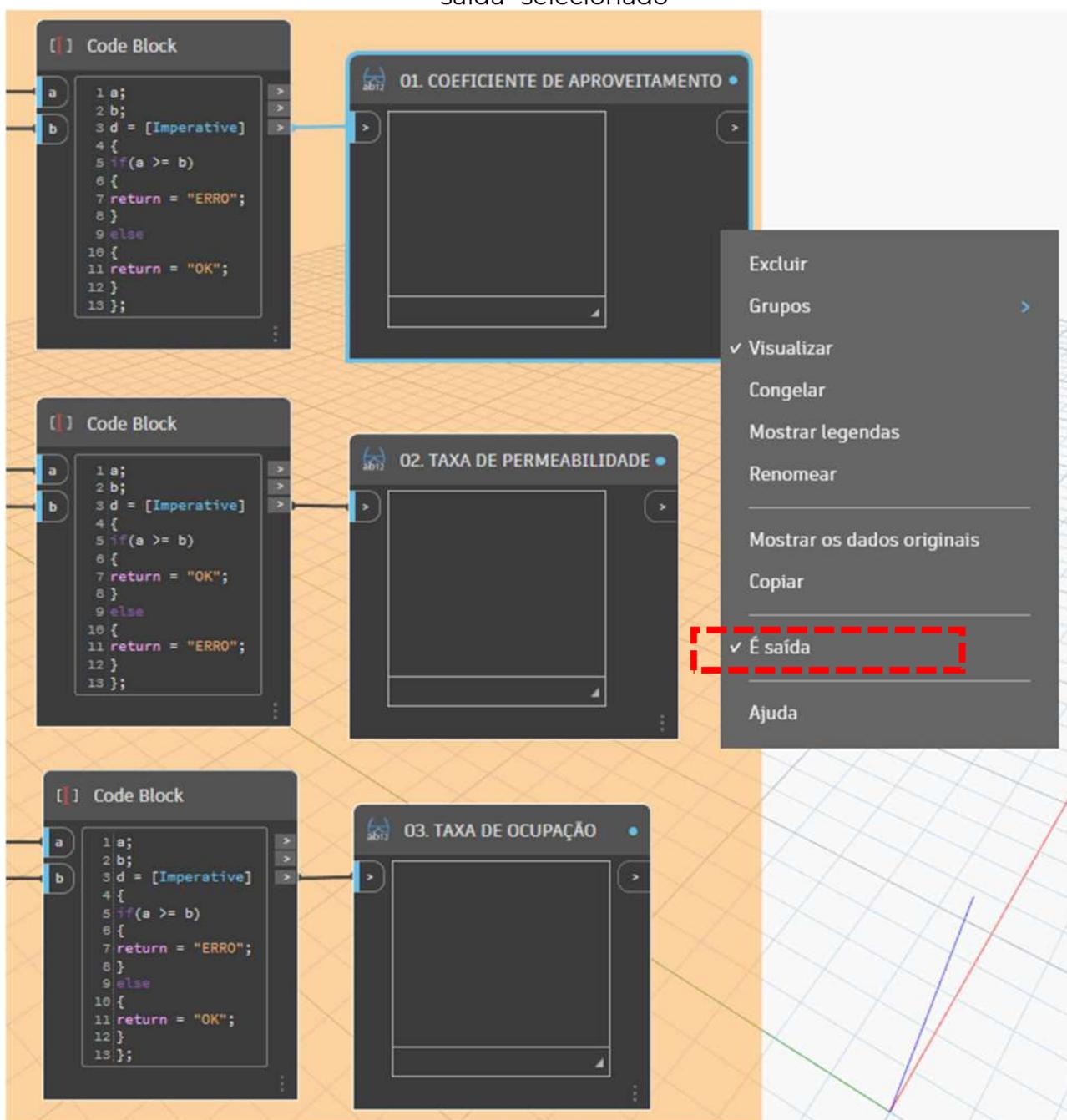


Fonte: Autor

Em seguida, os valores que popularam as listas são conflitadas com os valores impostos pela seleção do zoneamento. Através de um nó *code block* com uma condicional (*if-else*) são feitas as verificações finais. O resultado, após passagem pela última condicional é “OK” para o caso em que

os parâmetros de projeto encontram-se dentro dos parâmetros permitidos e “ERRO” para o contrário. Após isso, o valor “OK” ou “ERRO” é visualizado através de um nó *Watch* com a opção “É saída” devidamente selecionada. Isso irá fazer com que o usuário consiga visualizar o resultado, na interface nativa do Reprodutor do Dynamo, Figura 54.

Figura 54 Condicional da verificação numérica e nó *Watch* com “É saída” selecionado

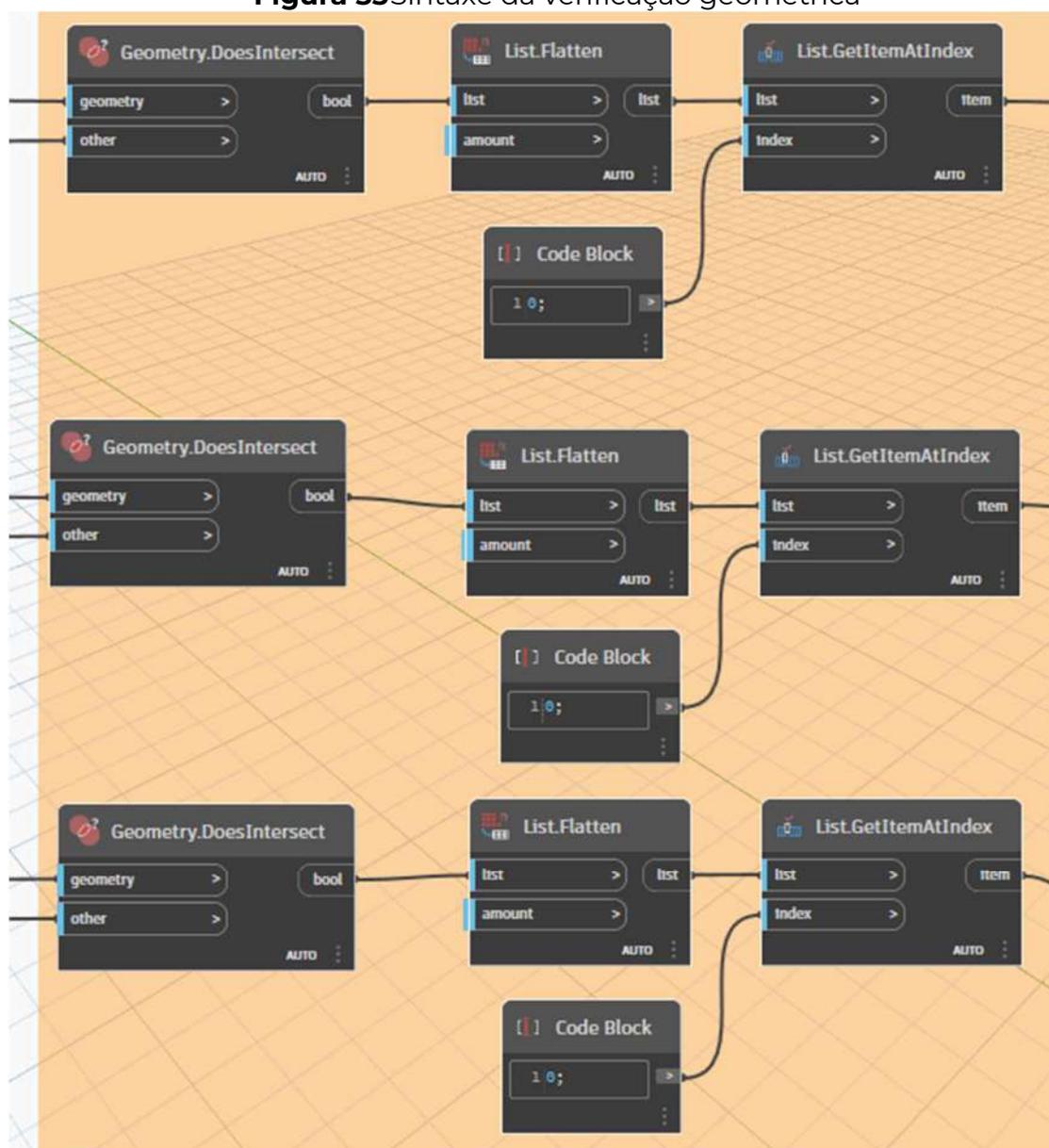


Fonte: Autor

- Verificação geométrica

Já as verificações geométricas de modelo, são feitas ao se conflitar o modelo de área construída, que em essência representa a edificação, com as demais geometrias transladadas e envelopamentos criados. São verificados de forma geométrica os seguintes parâmetros urbanísticos: (i) afastamentos laterais e de fundo; (ii) altura máxima da edificação; e (iii) recuo frontal. Tal verificação se dá através do nó *Geometry.DoesIntersect*, em que um resultado do tipo *True* ou *False* é obtido. Na sequência, o resultado também popula uma lista, Figura 55.

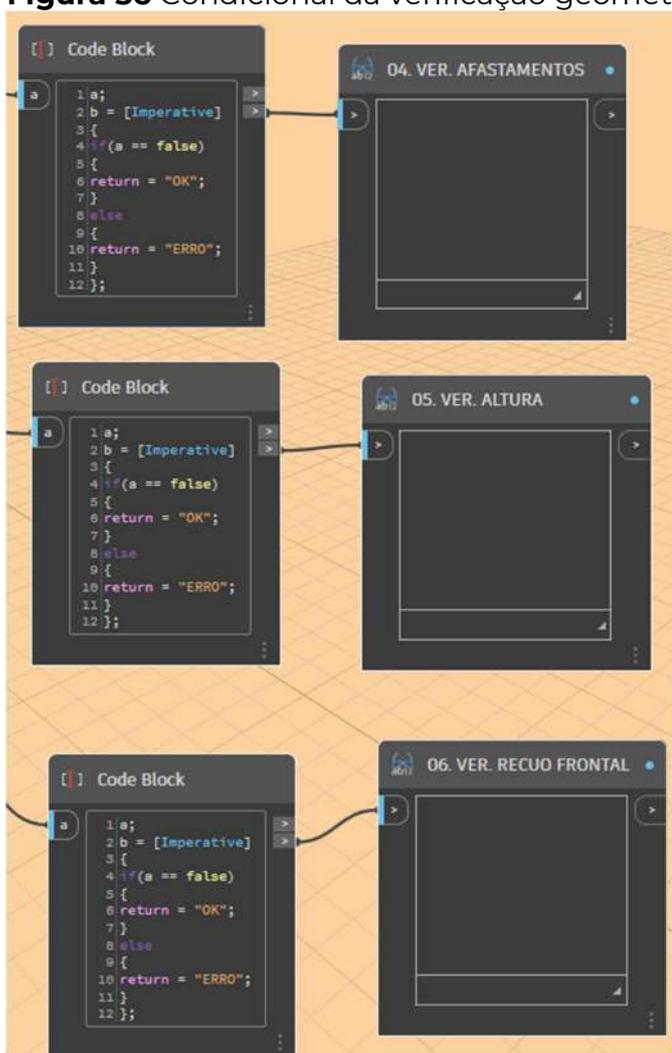
Figura 55 Síntaxe da verificação geométrica



Fonte: Autor

Os valores obtidos e que popularam as listas passam por uma última verificação, a fim de que o usuário também consiga identificar se tais parâmetros passaram ou não, no crivo do artefato. Da mesma forma que as verificações numéricas, as verificações geométricas são feitas com base em condicionais. O nó *Geometry.DoesIntersect* retorna um valor do tipo *False* para o caso em que não há intersecções entre as geometrias e um valor *True* para o caso em que há intersecção. Sendo assim, o nó *code block* populará um nó *Watch* com o valor “OK” para o caso em que há o retorno *False*, isto é, não há colisão; e “ERRO” para o caso em que há o retorno *True*, isto é, há colisão, Figura 56. Para que o resultado das verificações geométricas apareça ao usuário, as mesmas condições para os nós *Watch*'s, referentes às verificações numéricas devem ser aplicadas.

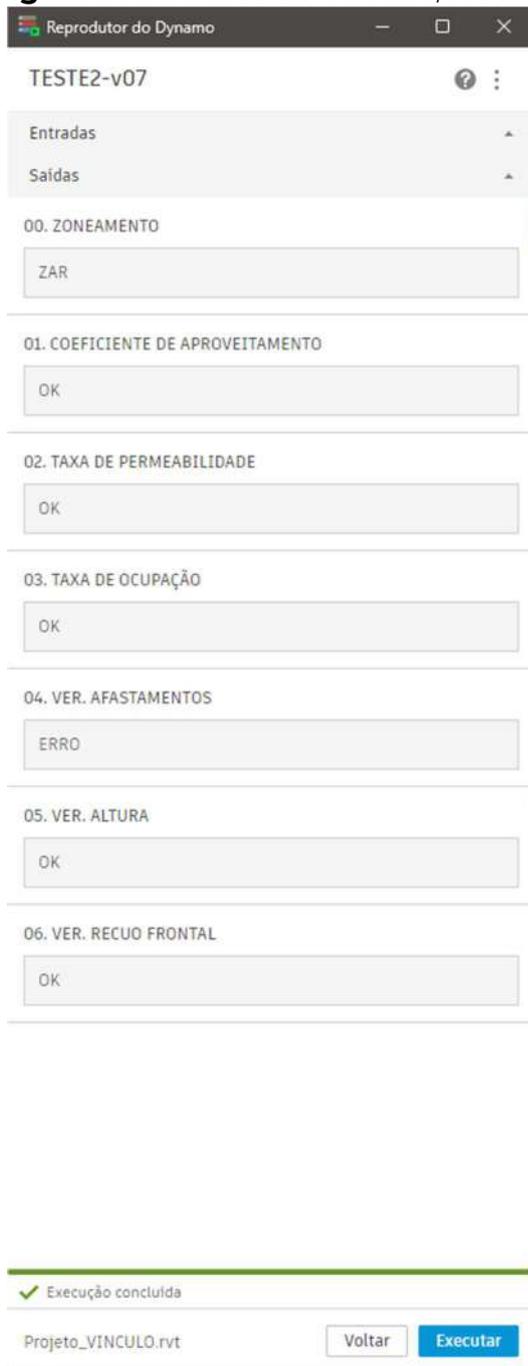
Figura 56 Condicional da verificação geométrica.



Fonte: Autor

Com isso, foi possível a obtenção da checagem dos parâmetros urbanísticos pelo artefato proposto. Todos os resultados, ao se executar a rotina do Dynamo, via Reprodutor Nativo aparecem ao usuário em forma de lista, sendo que cada linha, representa um nó *Watch* que possui a opção “É saída” devidamente selecionada Figura 57.

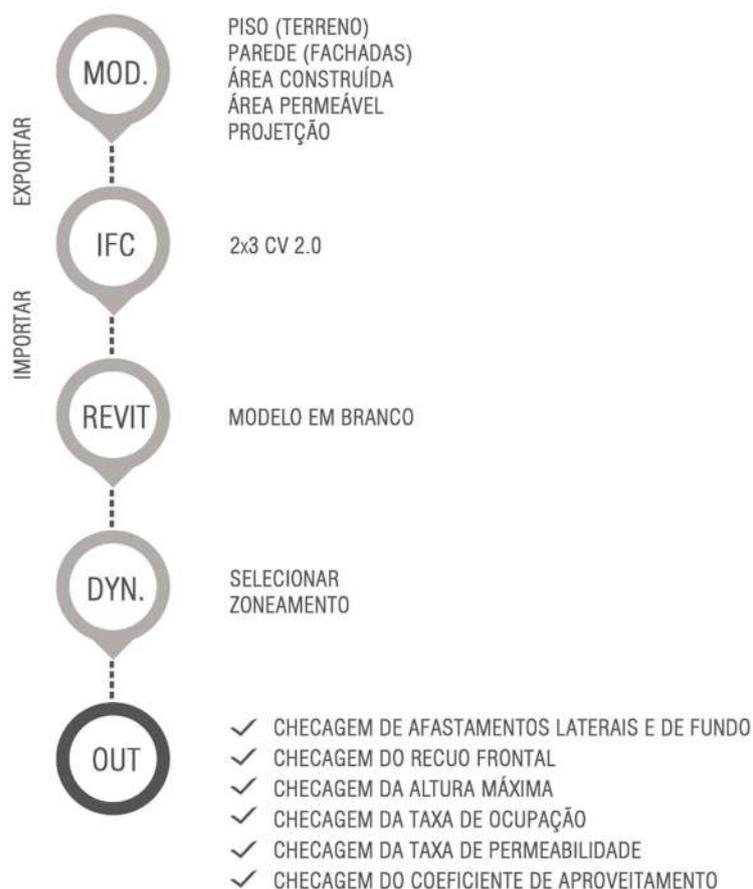
Figura 57 Interface do Usuário, resultados



Fonte: Autor

Tem-se então o seguinte fluxograma para o artefato final, Figura 58.

Figura 58 Implantação-fluxo do processo de validação de modelo BIM



Fonte: Autor

5.3 DIRETRIZES DE UTILIZAÇÃO

Para que o artefato possa operar de forma consistente e precisa, existem duas opções em relação à modelagem: (i) utilizar o *template* Revit desenvolvido para pesquisa ou (ii) utilizar nomenclatura específica para os elementos parede e piso. Em ambas as situações, deve-se:

- Criar um piso em toda a área do terreno. Esse piso deverá possuir o nome "TEMPLATE". Isso serve para determinar a área total do terreno e os afastamentos laterais e de fundo. É

recomendado que tal piso possua a mínima espessura o possível;

- Modelar as paredes externas, referentes às fachadas, de modo a conter o nome “TEMPLATE”. Isso serve para determinar se o recuo frontal está dentro dos limites impostos;
- Inserir plantas de áreas referentes às áreas construídas. Em Dados de Identidade, no item Nome, deve constar “ÁREA CONSTRUIDA” em todas as regiões pertinentes. Isso serve para verificação dos parâmetros de Coeficiente de Aproveitamento;
- Inserir plantas de áreas referentes às áreas permeáveis. Em Dados de Identidade, no item Nome, deve constar “ÁREA PERMEAVEL” em todas as regiões pertinentes. Isso serve para verificação do parâmetro de Taxa de Permeabilidade;
- Inserir planta de área referente à projeção. Em Dados de Identidade, no item Nome, deve constar “PROJEÇÃO” em todas as regiões pertinentes. Isso serve para verificação do parâmetro de Taxa de Ocupação;

Se os elementos a serem modelados - o piso representando o terreno e as paredes representando as fachadas - não puderem ser incluídos na documentação final, é recomendável que sejam ocultados do modelo. No entanto, ao exportar o modelo em formato IFC, é crucial que a exportação seja completa e não apenas dos elementos visíveis. Isso porque, com os elementos ocultos, eles não poderão ser importados no ambiente Dynamo.

Para execução da rotina Dynamo, deve-se seguir os seguintes passos:

- a. modelar o projeto, com as premissas previamente indicadas;
- b. exportar o modelo em formato IFC, considerando todos os objetos (mesmo os ocultos);

- c. importar o modelo IFC dentro de um arquivo em branco;
- d. rodar a rotina do Dynamo, via reprodutor nativo.

Com base nos resultados apresentados pelo artefato, na conclusão de sua execução, o projetista consegue de forma clara observar quais os parâmetros que porventura não obtiveram êxito nas análises. Em posse de tais informações, é possível que sejam realizadas as devidas alterações no projeto antes que o mesmo seja submetido para avaliação das diretrizes por parte da prefeitura.

6 AVALIAÇÕES

Serão apresentadas as avaliações encontradas na pesquisa, tanto do ponto de vista do artefato em si, isto é, da ferramenta desenvolvida, como dos desafios tecnológicos encontrados durante o seu desenvolvimento.

6.1 AVALIAÇÃO DO ARTEFATO

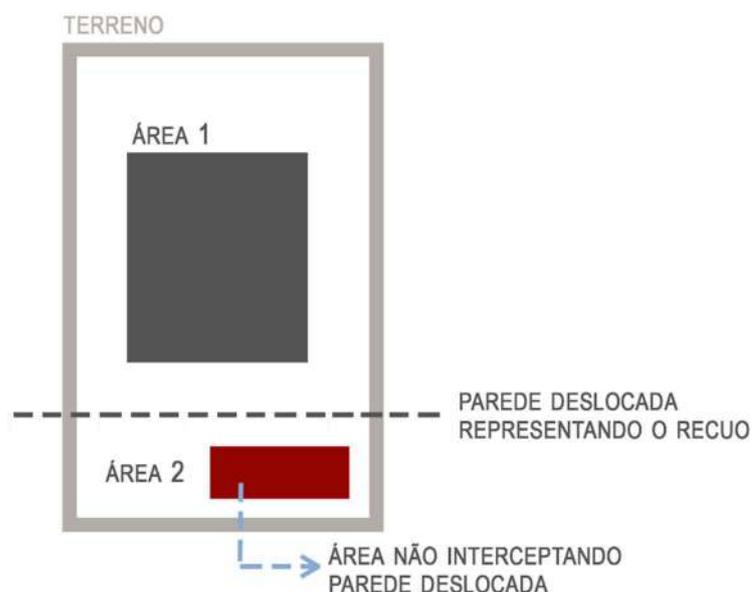
A partir da utilização do artefato com o objeto de estudo, foi possível avaliar a ferramenta quanto à confiabilidade das informações obtidas. A ferramenta foi desenvolvida dentro de limites e condições de operação que abrangem os seguintes parâmetros urbanísticos: afastamentos laterais e de fundo, recuo frontal, altura máxima permitida, área construída, área permeável, taxa de ocupação, coeficiente de aproveitamento e taxa de permeabilidade.

Assim que o usuário executa a rotina, a ferramenta fornece o resultado imediatamente. Na avaliação realizada, todos os parâmetros testados tiveram exatamente os mesmos valores obtidos em relação a uma avaliação analógica, ou seja, apresentou resultados efetivos.

Para aplicação desta ferramenta em outros projetos, deve-se levar em consideração alguns fatores. No objeto de estudo, o terreno possui quatro lados, sendo que a aresta frontal, que representa a fachada principal não é reta e sim um arco e o terreno considerado é plano. Para terrenos mais complexos, seja em relação às linhas limítrofes ou em relação ao seu desnível vertical, o artefato pode apresentar valores diferentes do esperado.

Já em relação à determinação do recuo frontal, a estratégia utilizada foi a de gerar um deslocamento de uma parede, que representa a fachada principal da edificação. Caso o projeto possua alguma área construída entre a parede original e a deslocada, o programa não identificará tal fator como erro e apresentará um resultado falso-positivo, Figura 59.

Figura 59 Demonstração de possível falha de reconhecimento de recuo frontal



Fonte: Autor

A LUOS da cidade de Poços de Caldas, utilizada como referência nessa pesquisa, traz várias cotas de afastamento lateral e de fundo, a depender da altura da edificação. Tal variável não foi considerada no desenvolvimento do artefato e, portanto, para edificações com altura que geraria distintos valores de afastamentos, seria constatado pelo artefato, um resultado equivocado.

Mais testes poderiam ser realizados, com condições adversas, tanto relacionadas ao terreno como à edificação em si, para uma avaliação mais abrangente. Também seria importante a validação dos arquivos IFC exportados de outros softwares de modelagem. Ainda, vale ressaltar que para a correta verificação do projeto pelo artefato, a modelagem pelo projetista deve seguir as diretrizes de utilização e constar com todas as áreas corretamente projetadas.

6.2 AVALIAÇÃO DOS DESAFIOS TECNOLÓGICOS

Os softwares tomados como base para o desenvolvimento do artefato foram o Revit em conjunto com o Dynamo, como demonstrado na

Proposição do Artefato. O Dynamo, possui uma interface intuitiva e uma comunidade ativa, o que auxiliou na troca de ideias e busca por pacotes de programação que facilitaram o desenvolvimento do artefato.

A análise das dificuldades tecnológicas encontradas, evidenciou que o tratamento de dados é um dos principais percalços encontrados pelos pesquisadores (HONGLING et al., 2016; LI; CAI; KAMAT, 2016; SEGHIER et al., 2017; FAROOQ; SHARMA; KUMAR, 2018; YUAN et al., 2019; GHANNAD et al., 2019; JIANG et al., 2019; LEE et al., 2019; AKANMU; OLAYIWOLA; ALFRED OLATUNJI, 2020; LAI et al., 2020; ALTINTAS; ILAL, 2021). Especialmente, no que se refere à transposição de dados de ordem natural ou linguagem natural (LN) para dados lógicos ou de linguagem de máquina (LM). Isso foi comprovado durante o desenvolvimento do artefato, uma vez que a LUOS utilizada como referência, considera um grande número de variáveis, o que torna a tarefa de transpor tais dados para um contexto lógico bastante árdua.

A utilização do esquema IFC como base para verificação automática de regras é uma estratégia adotada por diversos pesquisadores (HONGLING et al., 2016; LAI et al., 2020; SYDORA; STROULIA, 2020; YING; LEE, 2021). O seu uso traz uma neutralidade com relação ao software de modelagem, em que o usuário tem a liberdade de trabalhar em um programa que se sinta mais confortável, propiciando uma maior abrangência. Assim, esse esquema também foi utilizado para construção desse artefato. Entretanto, observa-se que é necessário realizar uma pré-checagem para averiguar se o modelo IFC contém as informações necessárias como propriedades, nomes e objetos para verificação automática na ferramenta, assim como em Hossain, Ahmed (2019) e Kim et al. (2020).

Em resumo, tendo em vista a complexidade das variáveis impostas pela LUOS, a utilização do Revit como base, o uso do esquema IFC e do Dynamo, conclui-se que a estratégia adotada apresentou diversas limitações e desafios tecnológicos, como: (i) a exigência do uso do Revit para funcionamento do artefato; (ii) a interpretação e manipulação dos objetos

importados para o ambiente do Dynamo; (iii) a inserção manual de informações no IFC que não eram importadas para o ambiente do Dynamo; (iv) a necessidade de verificar a modelagem BIM para evitar resultados falso-positivos; (v) o escopo restrito à uma tipologia de construção (habitação unifamiliar térrea); (vi) a transposição de LN para LM.

7 DISCUSSÃO E CONCLUSÃO

A partir da revisão da literatura e do desenvolvimento de um sistema de verificação automática de modelos BIM, tendo em vista a LUOS, buscou-se avaliar os desafios tecnológicos envolvidos no processo. Para tanto foi utilizada como referência a legislação da Prefeitura do Município de Poços de Caldas, MG.

Com a RSL, pode-se observar que existem diversas ferramentas utilizadas para o desenvolvimento de mecanismos que possibilitam a checagem automática de regras. Tais mecanismos dependem da abordagem adotada (*hard-coded*, orientada ao objeto, lógica e semântica/ontológica), sendo que a mais utilizada foi a orientada ao objeto. Os principais softwares para checagem automática de regras foram o *Dynamo VPL* para abordagem orientada ao objeto e o *Solibri Model Checker*, para abordagem *hard-coded*.

O uso de ferramentas de linguagem visual de programação torna-se um grande atrativo para o desenvolvimento de sistemas customizáveis. A estrutura de sua interface intuitiva, ampla biblioteca disponível e comunidade ativa, faz com que não seja necessário um amplo conhecimento de programação para a criação de ferramentas específicas.

A maioria das pesquisas analisadas fazem uso do esquema IFC para desenvolvimento de ferramentas de checagem automática de regras, o que tende a garantir uma maior abrangência. Contudo, o uso do esquema IFC não foi uma unanimidade por possuir limitações se comparado ao uso de formatos nativos em softwares proprietários, uma vez que esses permitem maior flexibilidade na manipulação de dados específicos.

Por meio dessa DSR, foi desenvolvido um artefato, tipo método e instanciação, para checagem de diversos parâmetros urbanísticos, tendo como público-alvo os projetistas. Para seu desenvolvimento foi utilizado o *Dynamo VPL* para programação. O artefato funciona na interface do Revit, que a partir de informações extraídas do esquema IFC, realiza as verificações programadas e, posteriormente, as exibe ao usuário.

O artefato desenvolvido, foi capaz de verificar os principais parâmetros urbanísticos utilizados na cidade de Poços de Caldas, como os afastamentos laterais e de fundo, recuo frontal, altura máxima da edificação e coeficientes de permeabilidade, ocupação e aproveitamento.

A utilização do artefato nas fases iniciais do projeto, especialmente durante os estudos preliminares, pode auxiliar os projetistas a lidarem com os principais parâmetros urbanísticos de maneira fácil, rápida e eficiente. Nesse momento, em que a concepção do projeto está sendo desenvolvida, diversos são os fatores que influenciam a sua evolução, dentre eles os parâmetros urbanísticos. Um sistema automatizado para uma checagem rápida dos principais parâmetros, pode otimizar essa tarefa, permitindo que o projetista se concentre em outras questões que não possam ser automatizadas.

O desenvolvimento de sistemas que permitam aos projetistas verificar automaticamente modelos BIM, a título de otimizar o processo de obtenção de aprovação por parte de prefeituras possui diversas complexidades envolvidas. Assim como em diversos estudos, umas das maiores dificuldades encontradas no desenvolvimento de ferramenta para checagem automática de regras residiu na manipulação de dados. Dificuldades como a transposição de textos em linguagem natural (leis, normas) para linguagem passível de ser lida por máquinas, foi um importante desafio tecnológico comprovado pelo desenvolvimento do artefato.

Foram observadas diversas limitações encontradas durante o desenvolvimento do artefato. A rotina foi desenvolvida com base na inferência de que parâmetros urbanísticos poderiam ser incorporados em ferramentas BIM para checagem automática de regras de modelos. Para melhoria da ferramenta, existem algumas questões que ainda precisam ser amadurecidas, como formato do terreno, distribuição das áreas construídas, altura da edificação em relação aos afastamentos laterais e de fundo.

A partir da RSL apresentada e do artefato criado, foi possível avaliar mais profundamente os desafios tecnológicos para o desenvolvimento de

um software para checagem automática de modelos BIM. Isso contribui para que as próximas iniciativas de criação desse tipo de software possam levar em consideração as dificuldades encontradas na idealização de um sistema mais abrangente. Tal sistema poderia auxiliar tanto os projetistas na pré-aprovação de projetos em BIM, quanto as prefeituras para avaliação dos projetos.

Futuramente a implementação de ajustes na rotina, como a inclusão de um conjunto de regras mais extensivo e a incorporação de coordenadas geográficas o uso do GIS para validação automática do zoneamento, poderia tornar viável o seu uso por parte do setor público municipal. Vale ressaltar que, para que esse tipo de sistema possa funcionar com a maior efetividade, seria interessante que as LUOS fossem estruturadas em conjunto com ferramentas de checagem automática de regras.

REFERÊNCIAS

- ADMIN, P. **Marionette Tutorials**. Disponível em: <https://forum.vectorworks.net/index.php?/articles.html/articles/marionette-tutorials-r432/>. Acesso em: 16 jul. 2022.
- AKANMU, A. A.; OLAYIWOLA, J.; ALFRED OLATUNJI, O. Automated checking of building component accessibility for maintenance. **Automation in Construction**, v. 114, 2020. DOI 10.1016/j.autcon.2020.103196.
- ALTIERI, A. **What is Marionette? A Look at Vectorworks' Algorithms-aided Design Tool**. Disponível em: <https://blog.vectorworks.net/what-is-marionette-a-look-at-vectorworks-algorithmic-modeling-tool>. Acesso em: 16 jul. 2022.
- ALTINTAS, Y. D.; ILAL, M. E. Loose coupling of GIS and BIM data models for automated compliance checking against zoning codes. **AUTOMATION IN CONSTRUCTION**, v. 128, 2021. ISSN 0926-5805. DOI 10.1016/j.autcon.2021.103743.
- AMOR, R.; DIMYADI, J. The promise of automated compliance checking. **Developments in the Built Environment**, v. 5, p. 100039, 2021. ISSN 2666-1659. DOI <https://doi.org/10.1016/j.dibe.2020.100039>.
- AUTODESK UNIVERSITY. **Unlock the Hidden Superpowers of Dynamo**. Disponível em: <https://www.autodesk.com/autodesk-university/article/Unlock-Hidden-Superpowers-Dynamo>. Acesso em: 16 jul. 2022.
- BAZJANAC, V. Model based cost and energy performance estimation during schematic design. **22nd Conference on Information Technology in Construction**, p. 677–688, 2005.
- BCA. **IFC**. Disponível em: <https://www1.bca.gov.sg/regulatory-info/building-control/corenet-x/resources/technical-knowledge-pages/ifc>. Acesso em: 9 jul. 2022.
- BEACH, T. H.; HIPPOLYTE, J.-L.; REZGUI, Y. Towards the adoption of automated regulatory compliance checking in the built environment. **AUTOMATION IN CONSTRUCTION**, v. 118, 2020. ISSN 0926-5805. DOI 10.1016/j.autcon.2020.103285.
- BEAZLEY, S.; HEFFERNAN, E.; MCCARTHY, T. J. Enhancing energy efficiency in residential buildings through the use of BIM: The case for embedding parameters during design. **Energy Procedia**, v. 121, p. 57–64, 2017. ISSN 1876-6102. DOI <https://doi.org/10.1016/j.egypro.2017.07.479>.

BLOCH, T.; SACKS, R. Comparing machine learning and rule-based inferencing for semantic enrichment of BIM models. **Automation in Construction**, v. 91, n. July 2017, p. 256–272, 2018. ISSN 09265805. DOI 10.1016/j.autcon.2018.03.018.

BLOCH, T.; SACKS, R. Clustering Information Types for Semantic Enrichment of Building Information Models to Support Automated Code Compliance Checking. **Journal of Computing in Civil Engineering**, v. 34, n. 6, p. 04020040, 2020. ISSN 0887-3801. DOI 10.1061/(asce)cp.1943-5487.0000922.

BRASIL. Lei de Licitações e Contratos Administrativos. . 2021.

BUILDING SMART INTERNATIONAL. **Industry Foundation Classes (IFC)**. Disponível em: <https://www.buildingsmart.org/standards/bsi-standards/industry-foundation-classes/>. Acesso em: 9 jul. 2022a.

BUILDING SMART INTERNATIONAL. **The importance of Industry Foundation Classes in Building Information Modelling**. Disponível em: <https://www.buildingsmart.org/the-importance-of-industry-foundation-classes-in-building-information-modelling/>. Acesso em: 9 jul. 2022b.

BUILDING SMART INTERNATIONAL. **Model View Definitions (MVD) - An Introduction**. Disponível em: <https://technical.buildingsmart.org/standards/ifc/mvd>. Acesso em: 9 jul. 2022c.

BURGGRÄF, P. et al. Creation of an expert system for design validation in BIM-based factory design through automatic checking of semantic information. **Procedia CIRP**, v. 99, p. 3–8, 2021. ISSN 2212-8271. DOI <https://doi.org/10.1016/j.procir.2021.03.012>.

CAMPINAS. **Formulários Cap - Aprovação De Projetos Construtivos, Habite-Se, Reforma E Demolição**. Disponível em: <https://www.campinas.sp.gov.br/governo/urbanismo/formularios.php>. Acesso em: 9 jul. 2022.

CHOI, J.; KIM, I. Development of rule-based building code compliance checking system for BIM-based quality improvement. **Information (Japan)**, v. 20, n. 4, p. 2929–2936, 2017. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85039788099&partnerID=40&md5=ed31cc7f6fb8776957ba8186e6e1c951>

CIRIBINI, A. L. C.; MASTROLEMBO VENTURA, S.; PANERONI, M. Implementation of an interoperable process to optimise design and construction phases of a residential building: A BIM Pilot Project. **Automation in Construction**, v. 71, p. 62–73, 2016. DOI 10.1016/j.autcon.2016.03.005.

CORENET. CORENET e-Information System. **CORENET**, 2016. Disponível em: <http://www.corenet.gov.sg/>. Acesso em: 9 jul. 2022.

DIMYADI, J. et al. Computerizing Regulatory Knowledge for Building Engineering Design. **JOURNAL OF COMPUTING IN CIVIL ENGINEERING**, v. 30, n. 5, 2016. ISSN 0887-3801. DOI 10.1061/(ASCE)CP.1943-5487.0000572.

DRESCH, A.; LACERDA, D. P.; ANTUNES JÚNIOR, J. A. V. **Design Science Research: método de pesquisa para avanço da ciência e tecnologia**. Porto Alegre: Bookman, 2015. ISBN 9783319106311.

DYNAMO BIM. **This is Dynamo**. Disponível em: <http://dynamobim.org/download/>. Acesso em: 16 jul. 2022.

DYNAMO PRIMER. **What is Dynamo?** Disponível em: https://primer.dynamobim.org/01_Introduction/1-2_what_is_dynamo.html. Acesso em: 16 jul. 2022.

EASTMAN, C. et al. Automatic rule-based checking of building designs. **Automation in Construction**, v. 18, n. 8, p. 1011–1033, 2009. ISSN 0926-5805. DOI <https://doi.org/10.1016/j.autcon.2009.07.002>.

FAN, S.-L. L.; CHI, H.-L. L.; PAN, P.-Q. Q. Rule checking Interface development between building information model and end user. **Automation in Construction**, v. 105, n. May, p. 102842, 2019. ISSN 09265805. DOI 10.1016/j.autcon.2019.102842.

FAROOQ, J.; SHARMA, P.; KUMAR, R. S. A BIM-based detailed electrical load estimation, costing and code checking. **International Journal of Electrical and Computer Engineering**, v. 8, n. 5, p. 3484–3495, 2018. DOI 10.11591/ijece.v8i5.pp3484-3495.

GETULI, V. et al. BIM-based Code Checking for Construction Health and Safety. **Procedia Engineering**, v. 196, n. June, p. 454–461, 2017. ISSN 18777058. DOI 10.1016/j.proeng.2017.07.224.

GHANNAD, P. et al. Automated BIM data validation integrating open-standard schema with visual programming language. **Advanced Engineering Informatics**, v. 40, p. 14–28, 2019. ISSN 1474-0346. DOI <https://doi.org/10.1016/j.aei.2019.01.006>.

GILL, T. G.; HEVNER, A. R. A fitness-utility model for design science research. **ACM Transactions on Management Information Systems**, v. 4, n. 2, 2013. ISSN 2158656X. DOI 10.1145/2499962.2499963.

GRAPHISOFT. **Solibri Model Checker**. Disponível em: <https://graphisoft.com/partner-solutions/solibri-model-checker>. Acesso em: 9 jul. 2022.

GRASSHOPPER3D. **Grasshopper - algorithmic modeling for Rhino**. Disponível em: <https://www.grasshopper3d.com/>. Acesso em: 16 jul. 2022.

GRUBER, T. R. Toward principles for the design of ontologies used for knowledge sharing? **International Journal of Human-Computer Studies**, v. 43, n. 5, p. 907–928, 1995. ISSN 1071-5819. DOI <https://doi.org/10.1006/ijhc.1995.1081>.

GUO, D.; ONSTEIN, E.; LA ROSA, A. D. An Approach of Automatic SPARQL Generation for BIM Data Extraction. **APPLIED SCIENCES-BASEL**, v. 10, n. 24, 2020. DOI 10.3390/app10248794.

GUO, D.; ONSTEIN, E.; LA ROSA, A. D. A Semantic Approach for Automated Rule Compliance Checking in Construction Industry. **IEEE Access**, 2021. DOI 10.1109/ACCESS.2021.3108226.

HJELSETH, E.; NISBET, N. N. Capturing normative constraints by use of the semantic mark-up RASE methodology. In: **Anais[...]**2011.

HOCHSCHEID, E.; HALIN, G. Micro BIM Adoption in Design Firms: Guidelines for Doing a BIM Implementation Plan. p. 864–871, 2019. DOI 10.3311/ccc2019-119.

HONGLING, G. et al. BIM and Safety Rules Based Automated Identification of Unsafe Design Factors in Construction. **Procedia Engineering**, v. 164, n. June, p. 467–472, 2016. ISSN 1877-7058. DOI <https://doi.org/10.1016/j.proeng.2016.11.646>. D

HOSSAIN, M. M.; AHMED, S. Developing an automated safety checking system using BIM: a case study in the Bangladeshi construction industry. **International Journal of Construction Management**, 2019. DOI 10.1080/15623599.2019.1686833.

ILAL, S. M.; GUNAYDIN, H. M. Computer representation of building codes for automated compliance checking. **AUTOMATION IN CONSTRUCTION**, v. 82, p. 43–58, 2017. ISSN 0926-5805. DOI 10.1016/j.autcon.2017.06.018.

JAIMES, M. **Entenda como funciona o programa Salvador 360 e seus 8 eixos de atuação direta**. Disponível em: <https://tvservidor.com.br/entenda-como-funciona-o-programa-salvador-360-e-seus-8-eixos-de-atuacao-direta/>. Acesso em: 10 mar. 2023.

JI, Y.; LEITE, F. Automated tower crane planning: leveraging 4-dimensional BIM and rule-based checking. **AUTOMATION IN CONSTRUCTION**, v. 93, p. 78–90, 2018. ISSN 0926-5805. DOI 10.1016/j.autcon.2018.05.003.

JIANG, S. et al. Combined MvdXML and Semantic Technologies for Green Construction Code Checking. **APPLIED SCIENCES-BASEL**, v. 9, n. 7, 2019. DOI 10.3390/app9071463.

JOTNE EPM TECHNOLOGY NEWS. **EDMmodelChecker**. Disponível em: <http://jotne.custompublish.com/edmmmodelchecker.78897.no.html>. Acesso em: 9 jul. 2022.

KANG, T. Rule-Based LEED Evaluation Method considering BIM Linkage and Variability. **KSCE Journal of Civil Engineering**, v. 24, n. 1, p. 110–121, 2020. DOI 10.1007/s12205-020-1026-x.

KHAN, N. et al. Excavation Safety Modeling Approach Using BIM and VPL. **Advances in Civil Engineering**, v. 2019, 2019. ISSN 16878094. DOI 10.1155/2019/1515808.

KIM, H. et al. Visual language approach to representing KBimCode-based Korea building code sentences for automated rule checking. **JOURNAL OF COMPUTATIONAL DESIGN AND ENGINEERING**, v. 6, n. 2, p. 143–148, 2019. DOI 10.1016/j.jcde.2018.08.002.

KIM, I. et al. Development of kbim e-submission prototypical system for the openbim-based building permit framework. **Journal of Civil Engineering and Management**, v. 26, n. 8, p. 744–756, 2020. ISSN 18223605. DOI 10.3846/jcem.2020.13756.

KINCELOVA, K. et al. Fire Safety in Tall Timber Building: A BIM-Based Automated Code-Checking Approach. **BUILDINGS**, v. 10, n. 7, jul. 2020. DOI 10.3390/buildings10070121.

KOO, B.; SHIN, B. Applying novelty detection to identify model element to IFC class misclassifications on architectural and infrastructure Building Information Models. **Journal of Computational Design and Engineering**, v. 5, n. 4, p. 391–400, 2018. ISSN 2288-4300. DOI <https://doi.org/10.1016/j.jcde.2018.03.002>.

LAI, H. et al. Implementation of building information modeling standards for the Shanghai Metro, China. **PROCEEDINGS OF THE INSTITUTION OF CIVIL ENGINEERS-MUNICIPAL ENGINEER**, v. 173, n. 3, p. 171–184, 2020. ISSN 0965-0903. DOI 10.1680/jmuen.18.00040.

LEE, H. et al. Translating building legislation into a computer-executable format for evaluating building permit requirements. **Automation in Construction**, v. 71, p. 49–61, 2016a. ISSN 09265805. DOI 10.1016/j.autcon.2016.04.008.

LEE, J. K. **Building Environment Rule and Analysis (BERA) Language**. 2011. Georgia Institute of Technology, 2011.

LEE, P. C. et al. An Efficient Design Support System based on Automatic Rule Checking and Case-based Reasoning. **KSCE Journal of Civil Engineering**, 2019. DOI 10.1007/s12205-019-1750-2.

LEE, W.-I. W.-I.; CHIANG, N.-C. N.-C. An Investigation of the Methods of Logicalizing the Code-Checking System for Architectural Design Review in New Taipei City. **Applied Sciences (Switzerland)**, v. 6, n. 12, 2016. ISSN 2076-3417. DOI 10.3390/app6120407.

LEE, Y.-C. **Rule Logic And Its Validation Framework Of Model View Definitions For Building Information Modeling**. 2015. Georgia Institute of Technology, 2015.

LEE, Y.-C. C.; SOLIHIN, W.; EASTMAN, C. M. The Mechanism and Challenges of Validating a Building Information Model regarding data exchange standards. **Automation in Construction**, v. 100, n. January, p. 118–128, 2019. ISSN 0926-5805. DOI <https://doi.org/10.1016/j.autcon.2018.12.025>.

LEE, Y.-C.; EASTMAN, C. M.; SOLIHIN, W. Logic for ensuring the data exchange integrity of building information models. **Automation in Construction**, v. 85, p. 249–262, 2018. ISSN 0926-5805. DOI <https://doi.org/10.1016/j.autcon.2017.08.010>.

LEE, Y. C. et al. Modularized rule-based validation of a BIM model pertaining to model views. **Automation in Construction**, v. 63, p. 1–11, 2016b. ISSN 09265805. DOI 10.1016/j.autcon.2015.11.006. Disponível em: <http://dx.doi.org/10.1016/j.autcon.2015.11.006>

LI, S.; CAI, H.; KAMAT, V. R. Integrating Natural Language Processing and Spatial Reasoning for Utility Compliance Checking. **JOURNAL OF CONSTRUCTION ENGINEERING AND MANAGEMENT**, v. 142, n. 12, 2016. ISSN 0733-9364. DOI 10.1061/(ASCE)CO.1943-7862.0001199.

MARCH, S. T.; SMITH, G. F. Design and natural science research on information technology. **Decision Support Systems**, v. 15, n. 4, p. 251–266, 1995. ISSN 01679236. DOI 10.1016/0167-9236(94)00041-2.

METROPOLIS SOLUTION. **Home - Metropolis Solutions**. Disponível em: <https://www.metropolis.solutions/>. Acesso em: 10 mar. 2023.

MORANDI, M. I. W. M.; CAMARGO, L. F. R. Revisão sistemática da literatura. In: **Design Science Research: Método de pesquisa para avanço da ciência e tecnologia**. São Paulo: Bookman, 2015. p. 204.

NACER, H.; AISSANI, D. Semantic web services: Standards, applications, challenges and solutions. **Journal of Network and Computer Applications**, v. 44, p. 134–151, 2014. ISSN 1084-8045. DOI <https://doi.org/10.1016/j.jnca.2014.04.015>.

NAWARI, N. Smartcodes and BIM. In: Structures Congress 2013: Bridging Your Passion with Your Profession - Proceedings of the 2013 Structures Congress, **Anais[...]**2013. DOI 10.1061/9780784412848.082.

NAWARI, N. O. A Generalized Adaptive Framework (GAF) for automating code compliance checking. **Buildings**, v. 9, n. 4, 2019. ISSN 20755309. DOI 10.3390/buildings9040086.

O'LEARY, M. L. **A first course in mathematical logic and set theory**. [s.l.] Hoboken, New Jersey : John Wiley & Sons, 2016.

PARK, S.; LEE, Y.-C.; LEE, J.-K. DEFINITION OF A DOMAIN-SPECIFIC LANGUAGE FOR KOREAN BUILDING ACT SENTENCES AS AN EXPLICIT COMPUTABLE FORM. **JOURNAL OF INFORMATION TECHNOLOGY IN CONSTRUCTION**, v. 21, p. 424–435, 2016. ISSN 1874-4753.

PAUWELS, P. et al. A performance benchmark over semantic rule checking approaches in construction industry. **ADVANCED ENGINEERING INFORMATICS**, v. 33, p. 68–88, 2017. ISSN 1474-0346. DOI 10.1016/j.aei.2017.05.001.

PREIDEL, C. et al. Towards code compliance checking on the basis of a visual programming language. **Journal of Information Technology in Construction**, v. 21, n. November, p. 402–421, 2016. ISSN 14006529.

PREIDEL, C.; DAUM, S.; BORRMANN, A. Data retrieval from building information models based on visual programming. **Visualization in Engineering**, v. 5, n. 1, 2017. DOI 10.1186/s40327-017-0055-0.

REVIT ADD-ONS. **SMARTreview APR – Automatic Code Compliance Checking**. Disponível em: <http://revitaddons.blogspot.com/2017/05/smartreview-apr-automatic-code.html>. Acesso em: 9 jul. 2022.

RHINOCEROS. **Grasshopper**. Disponível em: <https://www.rhino3d.com/7/new/#grasshopper>. Acesso em: 16 jul. 2022a.

RHINOCEROS. **Clash Detection**. Disponível em: <https://www.rhino3d.com/features/grasshopper/clash-detection/>. Acesso em: 16 jul. 2022b.

RIO DE JANEIRO. **Aprovação do Cadastro e Projeto da Obra**. Disponível em: <https://carioca.rio/servicos/aprovacao-do-cadastro-e-projeto-da-obra/>. Acesso em: 9 jul. 2022.

SACKS, R. et al. Semantic Enrichment for Building Information Modeling: Procedure for Compiling Inference Rules and Operators for Complex Geometry. **Journal of Computing in Civil Engineering**, v. 31, n. 6, p. 04017062, 2017. DOI doi:10.1061/(ASCE)CP.1943-5487.0000705.

SACKS, R. et al. Automating Design Review with Artificial Intelligence and BIM: State of the Art and Research Framework. In: **Anais[...]**2019. DOI 10.1061/9780784482421.045.

SACKS, R.; GIROLAMI, M.; BRILAKIS, I. Building Information Modelling, Artificial Intelligence and Construction Tech. **Developments in the Built Environment**, p. 100011, 2020a. ISSN 2666-1659. DOI <https://doi.org/10.1016/j.dibe.2020.100011>.

SACKS, R.; GIROLAMI, M.; BRILAKIS, I. Building Information Modelling, Artificial Intelligence and Construction Tech. **Developments in the Built Environment**, v. 4, n. May, p. 100011, 2020b. ISSN 2666-1659. DOI <https://doi.org/10.1016/j.dibe.2020.100011>.

SANTA CATARINA. **Caderno de especificações de projetos em BIM**. Disponível em: [https://www.sie.sc.gov.br/webdocs/sie/doc-tecnicos/labim/Caderno de Especificações de Projetos em BIM_102018.pdf](https://www.sie.sc.gov.br/webdocs/sie/doc-tecnicos/labim/Caderno%20de%20Especifica%C3%A7%C3%B5es%20de%20Projetos%20em%20BIM_102018.pdf). Acesso em: 18 ago. 2020.

SÃO PAULO. **Manual de Aprovação Eletrônica de Projetos**, 2022a. . Disponível em: <http://www3.prefeitura.sp.gov.br/sd2110/Forms/sisacoePH.aspx>. Acesso em: 7 jul. 2022.

SÃO PAULO. **Serviço de Licenciamento**. Disponível em: <https://slce.prefeitura.sp.gov.br/PaginasPublicas/SistemaIndisponivel.aspx>. Acesso em: 9 jul. 2022b.

SCHRANZ, C.; URBAN, H.; GERGER, A. Potentials of augmented reality in a bim based building submission process. **Journal of Information Technology in Construction**, v. 26, p. 441–457, 2021. DOI [10.36680/j.itcon.2021.024.85112277551&doi=10.36680%2Fj.itcon.2021.024&partnerID=40&md5=2ea5af6f880c4a8a48d21d27eb1a879d](https://doi.org/10.36680/j.itcon.2021.024.85112277551&doi=10.36680%2Fj.itcon.2021.024&partnerID=40&md5=2ea5af6f880c4a8a48d21d27eb1a879d)

SCHWABE, K.; TEIZER, J.; KÖNIG, M. Applying rule-based model-checking to construction site layout planning tasks. **Automation in Construction**, v. 97, n. November 2018, p. 205–219, 2019. ISSN 0926-5805. DOI <https://doi.org/10.1016/j.autcon.2018.10.012>.

SEDUR. **Salvador 360**. Disponível em: <http://www.sucom.ba.gov.br/noticias/sedur-apresenta-o-salvador-360-simplifica-para-especialistas-em-bim/>. Acesso em: 16 ago. 2020.

SEGHIER, T. E. et al. Building Envelope Thermal Performance Assessment Using Visual Programming and BIM, based on ETVV requirement of Green Mark and GreenRE. **INTERNATIONAL JOURNAL OF BUILT ENVIRONMENT AND SUSTAINABILITY**, v. 4, n. 3, p. 227–235, 2017. ISSN 1511-1369. DOI [10.11113/ijbes.v4.n3.216](https://doi.org/10.11113/ijbes.v4.n3.216).

SEGHIER, T. E.; AHMAD, M. H.; LIM, Y.-W. Automation of Concrete Usage Index (CUI) assessment using computational BIM. **INTERNATIONAL**

JOURNAL OF BUILT ENVIRONMENT AND SUSTAINABILITY, v. 6, n. 1, p. 23–30, jan. 2019. ISSN 1511-1369. DOI 10.11113/ijbes.v6.n1.319.

SHAHI, K.; MCCABE, B. Y.; SHAHI, A. Framework for Automated Model-Based e-Permitting System for Municipal Jurisdictions. **Journal of Management in Engineering**, v. 35, n. 6, 2019. DOI 10.1061/(ASCE)ME.1943-5479.0000712.

SHISHINA, D.; SERGEEV, P. REVIT|DYNAMO: Designing Objects of Complex Forms. Toolkit and Process Automation Features. **Architecture and Engineering**, v. 4, p. 30–38, 2019. DOI 10.23968/2500-0055-2019-4-3-30-38.

SINGAPORE GOVERNMENT. **CORENET e-Submission System**. Disponível em: <https://www.corenet.gov.sg/general/corenet-e-submission-system.aspx>. Acesso em: 9 jul. 2022.

SOLIBRI E-COMMERCE. **How to buy Solibri Subscription online**. Disponível em: <https://buy.solibri.com/>. Acesso em: 9 jul. 2022.

SOLIHIN, W. **A simplified BIM data representation using a relational database schema for an efficient rule checking system and its associated rule checking language**. 2016. Georgia Institute of Technology, 2016.

SOLIHIN, W. et al. Simplified schema queries for supporting BIM-based rule-checking applications. **Automation in Construction**, v. 117, n. May, p. 103248, 2020. ISSN 0926-5805. DOI <https://doi.org/10.1016/j.autcon.2020.103248>.

SOLIHIN, W.; EASTMAN, C. Classification of rules for automated BIM rule checking development. **Automation in Construction**, v. 53, p. 69–82, 2015a. ISSN 09265805. DOI 10.1016/j.autcon.2015.03.003.

SOLIHIN, W.; EASTMAN, C. M. Classification of rules for automated BIM rule checking development. **Automation in Construction**, v. 53, p. 69–82, 2015b. ISSN 09265805. DOI 10.1016/j.autcon.2015.03.003.

SOLIHIN, W.; EASTMAN, C. M. Classification of rules for automated BIM rule checking development. **AUTOMATION IN CONSTRUCTION**, v. 53, p. 69–82, 2015c. ISSN 0926-5805. DOI 10.1016/j.autcon.2015.03.003.

SOLIHIN, W.; EASTMAN, C. M. A knowledge representation approach in BIM rule requirement analysis using the conceptual graph. **Journal of Information Technology in Construction**, v. 21, p. 370–402, 2016. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85001090529&partnerID=40&md5=1dbe12994aeeb69c7003556b4b442240>

SOLIMAN-JUNIOR, J.; FORMOSO, C. T.; TZORTZOPOULOS, P. A semantic-based framework for automated rule checking in healthcare construction

projects. **CANADIAN JOURNAL OF CIVIL ENGINEERING**, v. 47, n. 2, SI, p. 202–214, 2020. ISSN 0315-1468. DOI 10.1139/cjce-2018-0460.

SOMAN, R. K.; MOLINA-SOLANA, M.; WHYTE, J. K. Linked-Data based Constraint-Checking (LDCC) to support look-ahead planning in construction. **Automation in Construction**, v. 120, p. 103369, 2020. ISSN 0926-5805. DOI <https://doi.org/10.1016/j.autcon.2020.103369>.

SONG, J. et al. Deep learning-based extraction of predicate-argument structure (PAS) in building design rule sentences. **Journal of Computational Design and Engineering**, v. 7, n. 5, p. 563–576, 2020. DOI 10.1093/jcde/qwaa046.

SYDORA, C.; STROULIA, E. Rule-based compliance checking and generative design for building interiors using BIM. **Automation in Construction**, v. 120, n. August, p. 103368, 2020. ISSN 09265805. DOI 10.1016/j.autcon.2020.103368.

TAKIM, R.; ZULKIFLI, M. H.; NAWAWI, A. H. Integration of Automated Safety Rule Checking (ASRC) System for Safety Planning BIM-Based Projects in Malaysia. **Procedia - Social and Behavioral Sciences**, v. 222, p. 103–110, 2016. ISSN 1877-0428. DOI <https://doi.org/10.1016/j.sbspro.2016.05.195>.

TORCH ENGENHARIA. **Manual de Boas Práticas**. [s.l.: s.n.].

TREBBI, C. et al. Clash Detection and Code Checking BIM Platform for the Italian Market. In: DANIOTTI, B.; GIANINETTO, M.; DELLA TORRE, S. (Ed.). **Digital Transformation of the Design, Construction and Management Processes of the Built Environment**. Cham: Springer International Publishing, 2020. p. 115–125. ISBN 978-3-030-33570-0. DOI 10.1007/978-3-030-33570-0_11.

TÜRKYILMAZ, E. A Method to Analyze the Living Spaces of Wheelchair Users Using IFC. **Procedia - Social and Behavioral Sciences**, v. 222, p. 458–464, 2016. ISSN 1877-0428. DOI <https://doi.org/10.1016/j.sbspro.2016.05.136>.

WARREN, D. R.; SALEEB, N. Improving Productivity by the Automation of Checking of 3D Parametric Modelling. **International Journal of Safety and Security Engineering**, v. 10, n. 4, p. 441–450, 2020. ISSN 2041904X. DOI 10.18280/ijssse.100402.

YING, H.; LEE, S. A rule-based system to automatically validate IFC second-level space boundaries for building energy analysis. **Automation in Construction**, v. 127, n. March, p. 103724, jul. 2021. ISSN 0926-5805. DOI <https://doi.org/10.1016/j.autcon.2021.103724>.

YOUNG-CHEOL, L.; EASTMAN, C. M.; SOLIHIN, W. Rules and validation processes for interoperable BIM data exchange. **Journal of Computational Design and Engineering**, v. 8, n. 1, p. 97–114, 2021. DOI 10.1093/jcde/qwaa064.

YUAN, J. et al. Accident prevention through design (PtD): Integration of building information modeling and PtD knowledge base. **AUTOMATION IN CONSTRUCTION**, v. 102, p. 86–104, jun. 2019. ISSN 0926-5805. DOI 10.1016/j.autcon.2019.02.015.

ZHANG, J.; EL-GOHARY, N. M. Extending Building Information Models Semiautomatically Using Semantic Natural Language Processing Techniques. **JOURNAL OF COMPUTING IN CIVIL ENGINEERING**, v. 30, n. 5, 2016. ISSN 0887-3801. DOI 10.1061/(ASCE)CP.1943-5487.0000536.

ZHANG, J.; EL-GOHARY, N. M. Integrating semantic NLP and logic reasoning into a unified system for fully-automated code checking. **AUTOMATION IN CONSTRUCTION**, v. 73, p. 45–57, jan. 2017. ISSN 0926-5805. DOI <https://doi.org/10.1016/j.autcon.2016.08.027>.

ZHONG, B. et al. Ontology-based framework for building environmental monitoring and compliance checking under BIM environment. **Building and Environment**, v. 141, n. December 2017, p. 127–142, 2018. ISSN 0360-1323. DOI <https://doi.org/10.1016/j.buildenv.2018.05.046>.

ZHONG, B. et al. Hyperledger fabric-based consortium blockchain for construction quality information management. **FRONTIERS OF ENGINEERING MANAGEMENT**, v. 7, n. 4, p. 512–527, 2020. DOI 10.1007/s42524-020-0128-y.

ZHONG, B. T. et al. Ontology-based semantic modeling of regulation constraint for automated construction quality compliance checking. **Automation in Construction**, v. 28, p. 58–70, 2012. ISSN 0926-5805. DOI <https://doi.org/10.1016/j.autcon.2012.06.006>.

ZHOU, P.; EL-GOHARY, N. Semantic information alignment of BIMs to computer-interpretable regulations using ontologies and deep learning. **Advanced Engineering Informatics**, v. 48, p. 101239, 2021. ISSN 1474-0346. DOI <https://doi.org/10.1016/j.aei.2020.101239>.

ZULKIFLI, M. H.; TAKIM, R.; NAWAWI, A. H. A proposed initial framework of ASRC system for BIM-based projects in Malaysia. **Jurnal Teknologi**, v. 78, n. 5–2, p. 61–67, 2016. DOI 10.11113/jt.v78.8491.

ANEXO A

Seção III, Dos Parâmetros Urbanísticos, de acordo com o Anexo II – Parâmetros Urbanísticos de Ocupação e Parcelamento do Solo – da lei complementar nº 92, de 05 de setembro de 2015 e subseções V – Do Recuo Frontal – e VI – Dos Afastamentos Laterais – da referida lei.

Zoneamento	Coefficiente de Aproveitamento (CA)	Tamanho mínimo de lote (m ²)	Taxa de Ocupação (%)	Taxa de Permeabilidade (%)	Altura máxima da edificação (m)	Testada mínima de lote (m)
ZPP	(A)	-	-	95	-	(G)
ZPAM ^(B)	0,10	20.000	0,5	90	-	(G)
ZPE-1	1,50	450	70	10	9,0	12
ZPE-2	2,0	300	80	10	12,0	12
ZPE-3	3,0	360	80	10	16,0 ^(C)	12
ZAR	1,2	450 ^(D)	50	30	16,0	12
ZAM	3,0	300	80	10	16,0	12
ZAP	6,5	250	85	10	48,0	10
ZEIS-1	(A)					
ZEIS-2	Parâmetros da zona mais permissível do entorno imediato					
ZI ^(E)	2,0	2.000 ^(F)	70	10	-	20
ZRPA	-	-	10	90	-	(G)
ZR	-	20.000 ^(F)	30	80	-	(G)

(A) a ser definido caso a caso.

(B) observar o disposto no parágrafo 4º do artigo 5º desta lei complementar.

(C) não incluída a altura da caixa d'água e casa de máquinas e dependências do apartamento imediatamente abaixo, limitado a um pavimento e desde que possua acesso dependente, com área construída de no máximo 50% da área do pavimento tipo, quando for o caso.

(D) no o local denominado "Fazenda das Árvores", de que trata o Art. 10-M da Lei Complementar Nº 74/06, o tamanho mínimo de lote será de 600,00m² (seiscentos metros quadrados).

(E) Os parâmetros estabelecidos poderão ser flexibilizados para os lotes aprovados antes da vigência desta Lei complementar, mediante justificativa circunstanciada que comprove sua inviabilidade, e manifestação favorável do órgão de planejamento do Município, ouvido o COMDURT.

(F) Tamanho mínimo de gleba, estabelecido pelo INCRA, admitidas, exclusivamente, as atividades previstas no parágrafo 12 do artigo 5º desta lei complementar.

(G) A relação entre profundidade e testada dos lotes não poderá ter coeficiente superior a cinco.

(H) Exceto para os lotes localizados na Rua Mucovita, no trecho compreendido entre a Rodovia Geraldo Martins Costa (Rodovia do Contorno) e o Ribeirão das Várzeas, de propriedade do Município, onde as áreas mínimas dos lotes não poderão ser inferiores a 600,00m² (seiscentos metros quadrados) e testadas mínimas de 15,00m (quinze metros).

Art. 13. Considera-se Recuo Frontal mínimo das edificações – RF, a menor distância permitida entre a edificação e o alinhamento do terreno, medida perpendicular a este, sendo de 3m (três metros) para todas as Zonas.

§ 1º. O recuo frontal poderá ser dispensado até a altura de 6m (seis metros) para edificações a serem implantadas em face de quadras que apresentem no mínimo 40% (quarenta por cento) dos lotes comprovadamente aprovados após a vigência da Lei Complementar n. 74/2006, com edificações verticais com, no mínimo, três pavimentos ou 8m (oito metros) de altura, construídas no alinhamento.

§ 2º. No caso de edificação a ser executada após demolição de outra, comprovadamente existente até a vigência da Lei Complementar n. 74/2006, o recuo frontal poderá ser ocupado na mesma proporção que anteriormente ocupava junto ao alinhamento predial, até a altura de 6m (seis metros), excluído o telhado.

§ 3º. Nos lotes oriundos de parcelamentos anteriores à publicação da Lei Complementar n. 74/2006, cuja inclinação natural seja igual ou superior a 50% (cinquenta por cento), poderão ser executadas garagens no alinhamento frontal, não podendo exceder à metade da testada do lote.

§ 4º. A inclinação de que trata o § 3º deste artigo, será calculada para uma profundidade de 5m (cinco metros), a partir do alinhamento do terreno.

§ 5º. Para edificações em vias estruturais projetadas a área *non aedificandi* frontal mínima será de 31 m (trinta e um metros), medidos a partir do eixo da via projetada, de onde será computado o recuo frontal da edificação.

§ 6º. Para edificações em vias estruturais já implantadas, a área *non aedificandi* frontal mínima será de 15m (quinze metros), contados da borda externa da via para fins de implantação de via paralela.

§ 7º. Somente serão dispensadas de área *non aedificandi* as vias estruturais em que não houver viabilidade técnica de implantação, mediante parecer favorável da Secretaria de Planejamento e Coordenação.

§ 8º. O recuo frontal poderá ser dispensado até a altura $H = 6m$ (seis metros) excluído o telhado, em lotes comprovadamente aprovados antes da publicação da Lei Complementar n. 74/2006, desde que atendidas as seguintes condições:

- I- nos casos em que 50% (cinquenta por cento) dos lotes da face de quadra apresentem edificações construídas no alinhamento ou com recuos inferiores aos permitidos nesta lei complementar, sendo que:
 - a) quando a quadra for dividida por uma via pública oficial, as quadras resultantes serão consideradas distintas, não sendo permitido o somatório de seus lotes;

- b) quando existirem lotes vagos na face da quadra, estes serão considerados recuados;
 - c) no caso de regularização de edificações construídas no alinhamento ou que apresentem recuos inferiores ao determinado por esta lei complementar, deverão ser computadas como alinhadas;
 - d) para efeito de contagem de lotes, será adotada a regra do arredondamento universal;
- II- nos casos de altura (H) maior de 6m (seis metros), o restante da edificação deverá obedecer ao recuo frontal mínimo de 3m (três metros).

Art. 14. Será admitida a construção de guarita na área delimitada pelo recuo frontal, desde que apresente área máxima construída de 7m² (sete metros quadrados).

Art. 15. Os afastamentos mínimos laterais e de fundo das edificações são os seguintes:

- I- 1,50m (um metro e cinquenta centímetros) para as edificações com altura – H, menor que 6 m (seis metros), excluído o telhado;
- II- 2 m (dois metros) para as edificações com H maior ou igual a 6 m (seis metros) e menor ou igual a 15 m (quinze metros);
- III- os obtidos pela aplicação da fórmula constante do Anexo III desta lei complementar, para as edificações com H maior que 15 m (quinze metros).

§ 1º. Entende-se por H a distância vertical, em metros, medida entre o ponto médio do passeio lindeiro à testada do lote, até o ponto mais alto da edificação, incluindo todos os seus elementos construtivos, à exceção do previsto no § 5º do art. 11 desta lei complementar.

§ 2º. Os subsolos somente poderão ser utilizados como área útil até a cota máxima de 9m (nove metros), sendo esta referência a distância vertical entre a cotada 1m (um metro) acima do ponto médio do passeio lindeiro à testada do lote, até o piso do último pavimento inferior, não sendo considerados para o cálculo do H, sendo que:

- I- caso a cota máxima seja ultrapassada, a altura total será considerada para o cálculo do H;
- II- os afastamentos laterais e de fundo deverão atender à

mesma projeção do pavimento imediatamente superior ao ponto médio do passeio lindeiro à testada do lote, exceto para garagem no subsolo, desde que garantidas as condições de iluminação e ventilação adequadas ao uso.

§ 3°. Para terrenos em aclave ou declive, o H poderá ser considerado pelo ponto médio do terreno onde se inicia o plano de fachada da edificação.

§ 4°. Para os lotes que tenham sido, comprovadamente, aprovados antes da vigência da Lei Complementar n. 74/2006, poderão ser dispensados os afastamentos laterais e de fundos mínimos para edificações com H inferior a 6 m (seis metros), excluído o telhado, nos seguintes casos:

- I- lotes com testada menor ou igual a 10 m (dez metros);
- II- lotes com largura média menor ou igual a 10 m (dez metros);
- III- lotes com área inferior a 200m² (duzentos metros quadrados);
- IV- quando não houver abertura nas divisas, prevalecendo, se houver, o afastamento mínimo de 1,50m (um metro e cinquenta centímetros).

§ 5°. Para os lotes que tenham sido, comprovadamente, aprovados antes da vigência da Lei Complementar n. 74/2006, para os localizados em ZAP – Zona de Adensamento Preferencial e para os lotes de esquina, poderá ser dispensado o afastamento lateral mínimo em um dos lados para edificações com H inferior a 6m (seis metros) excluído o telhado, nos seguintes casos:

- I- lotes com testada maior que 10 m (dez metros) e menor ou igual a 12m (doze metros);
- II- lotes com largura média menor ou igual a 12m (doze metros);
- III- lotes de esquina, comprovadamente aprovados antes da publicação desta lei complementar;
- IV- para os lotes em ZAP que possuam testada igual a 10m até 12m;
- V- quando não houver abertura nas divisas, prevalecendo, se houver, o afastamento mínimo de 1,50 m (um metro e cinquenta centímetros).

§ 6°. A largura média de que tratam os § 4° e 5° deste artigo, é obtida pela média aritmética das larguras do lote, tomadas a cada 1m (um metro), a partir da testada até o fundo do lote.

§ 7°. Os afastamentos laterais e de fundo poderão ser dispensados para implantação de edículas destinadas a serviços de apoio ao uso residencial, sendo sua área construída computada no cálculo dos demais parâmetros previstos no Anexo II desta lei complementar e devendo estar a, no mínimo, 1,50m (um metro e meio) de distância das demais edificações existentes ou a construir no lote, nas seguintes condições:

- I- limitada a um pavimento;
- II- área construída não superior a 20% (vinte por cento) da área do lote;
- III- não poderá ser utilizada como residência permanente.

Art. 16. No caso de edificação constituída de dois ou mais blocos independentes, a distância entre eles deverá obedecer ao dobro dos afastamentos mínimos laterais e de fundo previstos nesta lei complementar, não podendo ser inferior a 1,50m (um metro e cinquenta centímetros).

Parágrafo único. No caso de bloco único, a distância mínima entre unidades deverá ser de 1,50 (um metro e cinquenta centímetros) para aberturas existentes em uma das faces e de 3m (três metros) para abertura em ambas as faces.

Art. 17. Para os lotes aprovados à época anterior à vigência da Lei Complementar n. 74/2006, o afastamento lateral e de fundo, para cada uma das divisas, poderá ser dispensado na ZAP e nas ZPEs, desde que atendidas as seguintes condições:

- I- existência de edificação com empena cega na divisa dos respectivos lotes vizinhos;
- II- altura máxima da edificação na divisa limitada a 6m (seis metros), acima da cota do terreno natural, excluído o telhado.

§ 1°. Ocorrendo o disposto no caput deste artigo, será admitida a superação da taxa de ocupação prevista para a Zona.

§ 2°. É proibida a construção sem afastamentos laterais e de fundo nas partes das edificações nas quais haja aberturas voltadas para as divisas laterais ou de fundo.

ANEXO B

Da Apresentação de Projetos, de acordo com os artigos 2º e 3º, do capítulo I, do decreto nº 11.687, de 05 de setembro de 2015.

Art. 2º O interessado em solicitar a aprovação de projetos de construção e regularização em áreas urbanas deverá apresentar a seguinte documentação:

- I. quita de arrecadação das taxas municipais (original ou cópia);
- II. requerimento padrão, fornecido pelo setor de Protocolo e disponibilizado no site eletrônico da Prefeitura, assinado pelo responsável técnico (RT) e/ou proprietário (Anexo I);
- III. uma cópia do projeto completo, assinado pelo autor do projeto e responsável técnico;
- IV. certidão da Matrícula de Registro do Imóvel (RI), constando as informações atuais, e cópia do contrato de compra e venda do terreno e/ou escritura, nos casos em que a matrícula do imóvel não esteja no nome do proponente;
- V. comprovante de quitação do ISSQN do RT.

§ 1º. É direito do possuidor requerer, perante a Prefeitura, licença para realizar obras e edificações no imóvel, desde que expressamente autorizado pelo proprietário para tal finalidade, mediante a apresentação de qualquer dos seguintes documentos:

- I. compromisso de compra e venda;
- II. certidão do Registro Imobiliário contendo as características do imóvel, quando o requerente possuir escritura definitiva sem registro ou quando for possuidor "ad usucapionem" com ou sem justo título ou ação em andamento;
- III. opção de compra com respectivas assinaturas reconhecidas em cartório, acompanhada pelo documento que caracterize o possuidor e o proprietário do imóvel.

§ 2º. No caso de proprietário representado por procurador, deverá ser apresentada procuração com firma reconhecida em cartório e cópias autenticadas dos documentos de identidade com foto.

§ 3º. Nos casos de espólio, deverá ser apresentada carta de nomeação do inventariante expedida pelo Juiz de Direito.

Art. 3º. Os projetos a serem submetidos a apreciação deverão ser apresentados em forma de projeto completo, constando obrigatoriamente:

- I. planta de locação em escala legível com a indicação de:
 - a) todos os logradouros limitrofes;
 - b) todos os elementos que definem a forma e as dimensões do lote e das edificações, inclusive as projeções de seus pavimentos e a posição destas no lote com todos os afastamentos e recuos;
 - c) indicação de distâncias entre edificações no mesmo lote, quando necessário;
 - d) locação das fossas sépticas e filtros anaeróbios, dos cursos d'água e a distância das margens destes às edificações, quando necessário;
 - e) indicação da área permeável com todas as suas dimensões e/ou a locação e o dimensionamento do sistema de captação e drenagem, quando necessário;
 - f) indicação das curvas de nível a cada metro de desnível, ou cotas de nível das extremidades do lote;
- II. plantas baixas de todos os pavimentos (quando se tratar de pavimento tipo, deverá ser indicado a quantidade), em escala legível que indiquem:
 - a) todas as dimensões externas;
 - b) recuos e afastamentos devidamente cotados;
 - c) os acessos à(s) unidade(s);
 - d) vagas de garagem devidamente cotadas;
 - e) indicação do(s) uso(s);
- III. um corte esquemático que contenha:
 - a) identificação dos pavimentos;
 - b) a cota da distância entre o ponto mais baixo e o ponto mais alto da edificação;
 - c) a cota da distância entre o ponto médio do meio fio (PMMF) e a altura máxima da edificação;
 - d) o perfil natural do terreno;
 - e) a cota da distância entre o ponto médio do plano de fachada e a altura máxima da edificação e a cota da distância entre o ponto médio do plano de fachada e o perfil natural do terreno;

- f) a cota da altura do subsolo, quando houver;
- IV. planta de cobertura, em escala legível, que indique o sentido das águas, as calhas, os rufos e as cotas das dimensões dos beirais;
- V. detalhe da calçada em escala 1:100, obedecendo a legislação vigente, contendo:
 - a) desenho da calçada com cota da largura, indicação do meio fio e alinhamento predial, devidamente cotados e com indicação da inclinação transversal;
 - b) indicação de equipamentos urbanos existentes (postes, telefones públicos, lixeiras, árvores, entre outros);
 - c) locação da rampa de acesso à garagem, devidamente cotada, respeitando as normas municipais;
 - d) locação da área permeável devidamente cotada;
- VI. carimbo padrão, conforme Anexo III, devidamente preenchido, com a planta de situação do imóvel podendo ser sem escala, que contenha: a indicação do Norte, a distância a uma rua com denominação oficial ou designação notória e a denominação das ruas de acesso;
- VII. relatório fotográfico do lote ou da edificação, quando se tratar de projeto de regularização ou intervenção em edificações existentes (Anexo IV);
- VIII. quadro de áreas de unidades autônomas, quando houver mais de uma unidade e tratar-se de regularização;
- IX. memorial de cálculo das áreas de projeção por pavimento.

ANEXO C

Dos Procedimentos Internos, de acordo com o artigo 7º, do capítulo II, do decreto nº 11.687, de 05 de setembro de 2015.

Art. 7º. Os processos seguirão os seguintes

trâmites:

- I. o Setor de Protocolo encaminhará o processo à Divisão de Controle de Parcelamento e Uso do Solo (DCPUS), que procederá a análise, podendo encaminhá-lo ao Departamento de Meio Ambiente (DMA), à Divisão de Patrimônio Construído e Tombamento (DPCT) ou à Divisão de Planejamento (DP), quando o técnico analista julgar necessário;
- II. nos casos em que o processo for encaminhado aos departamentos mencionados no inciso I, estes deverão retornar o parecer conclusivo à DCPUS em até 10 (dez) dias úteis;
- III. a DCPUS informará ao RT o resultado da análise, através de Relatório de Avaliação Técnica (Anexo V), em até 15 (quinze) dias úteis se a análise se restringir ao departamento e em até 25 (vinte e cinco) dias úteis se a análise envolver outros departamentos;
- IV. as correções apontadas no Relatório de Avaliação Técnica deverão ser apresentadas pelo requerente em até 60 (sessenta) dias úteis;
- V. protocoladas as correções solicitadas, o processo será encaminhado à DCPUS para nova análise, podendo ser realizadas consultas ao DMA, DPCT e DP, quando expresso no primeiro Relatório de Avaliação Técnica, retornando ao RT em até 15 (quinze) dias úteis;
- VI. não atendidas as correções na íntegra, o requerente será notificado novamente, estabelecendo-se novo prazo de 20 (vinte) dias úteis para o atendimento, não devendo, nesta fase, ser acrescentado nenhum fato novo às exigências iniciais, exceto se for constatada alguma irregularidade decorrente de alterações no projeto reapresentado;
- VII. findo este prazo, sem o atendimento às correções ou justificativa, o processo será indeferido;
- VIII. atendidas todas as normas exigíveis e não havendo pendências, será solicitada a apresentação das cópias para aprovação do projeto e cópia da Anotação de Responsabilidade Técnica (ART) comprovando o registro do projeto no Conselho Regional de Engenharia e Agronomia (CREA), ou Relatório de Responsabilidade Técnica (RRT) do Conselho de Arquitetura e Urbanismo (CAU), com os respectivos comprovantes de pagamento.