

INSTITUTO FEDERAL

Goiás

Câmpus Formosa

CURSO SUPERIOR DE BACHARELADO EM ENGENHARIA CIVIL

AUTOMAÇÃO DE ANÁLISES DE PROJETO NO AUTODESK REVIT

JOÃO VICTOR DE OLIVEIRA AGUIAR

Formosa – GO

2022



INSTITUTO FEDERAL
Goiás

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
SISTEMA INTEGRADO DE BIBLIOTECAS

**TERMO DE AUTORIZAÇÃO PARA DISPONIBILIZAÇÃO
NO REPOSITÓRIO DIGITAL DO IFG - ReDi IFG**

Com base no disposto na Lei Federal nº 9.610/98, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia de Goiás, a disponibilizar gratuitamente o documento no Repositório Digital (ReDi IFG), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, em formato digital para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IFG.

Identificação da Produção Técnico-Científica

- | | |
|--|---|
| ☐ Tese | ☐ Artigo Científico |
| ☐ Dissertação | ☐ Capítulo de Livro |
| ☐ Monografia – Especialização | ☐ Livro |
| ☒ TCC - Graduação | ☐ Trabalho Apresentado em Evento |
| ☐ Produto Técnico e Educacional - Tipo: _____ | |

Nome Completo do Autor: João Victor de Oliveira Aguiar

Matrícula: 20161070080027

Título do Trabalho: Automação de análises de projeto no Autodesk Revit.

Autorização - Marque uma das opções

1. ☒ Autorizo disponibilizar meu trabalho no Repositório Digital do IFG (acesso aberto);
2. ☐ Autorizo disponibilizar meu trabalho no Repositório Digital do IFG somente após a data ____/____/____ (Embargo);
3. ☐ Não autorizo disponibilizar meu trabalho no Repositório Digital do IFG (acesso restrito).

Ao indicar a opção **2 ou 3**, marque a justificativa:

- ☐ O documento está sujeito a registro de patente.
☐ O documento pode vir a ser publicado como livro, capítulo de livro ou artigo.
☐ Outra justificativa: _____

DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA

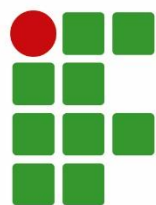
O/A referido/a autor/a declara que:

- i. o documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;
- ii. obteve autorização de quaisquer materiais inclusos no documento do qual não detém os direitos de autor/a, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia de Goiás os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;
- iii. cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia de Goiás.

Formosa-GO, 04/04/2022.

Local Data

Assinatura do Autor e/ou Detentor dos Direitos Autorais



INSTITUTO FEDERAL

Goiás

Câmpus Formosa

CURSO SUPERIOR DE BACHARELADO EM ENGENHARIA CIVIL

JOÃO VICTOR DE OLIVEIRA AGUIAR

AUTOMAÇÃO DE ANÁLISES DE PROJETO NO AUTODESK REVIT

Trabalho de Conclusão de Curso submetido à Coordenação do Curso de Bacharelado em Engenharia Civil do Instituto Federal de Educação, Ciência e Tecnologia de Goiás, Campus Formosa, como parte dos requisitos para a obtenção do grau de Bacharel em Engenharia Civil.

Orientador: Prof. Me. Vitor Amadeu da Silva Feitoza

Formosa – GO

2022

Dados Internacionais de Catalogação na Publicação

A283a

Aguiar, João Victor de Oliveira

Automação de análises de projeto no Autodesk Revit / João Victor de Oliveira Aguiar - Formosa, 2022.

122 f.: il ; 30 cm.

Orientador: : Prof. Me. Vitor Amadeu da Silva Feitoza

Monografia (graduação) – Bacharelado em Engenharia Civil, Instituto Federal de Educação, Ciência e Tecnologia de Goiás. Formosa, 2022.

1. Construção civil - Autodesk Revit. 2. Building Information Modeling (BIM). 3. Application Programming Interface (API). I. Feitoza, Vitor Amadeu da Silva. II. Título.

CDD: 692.068



INSTITUTO FEDERAL
Goiás

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE GOIÁS
CÂMPUS FORMOSA

ATA Nº 07/ 2022

Engenharia Civil / Câmpus Formosa

ATA DA SESSÃO PÚBLICA DE APRESENTAÇÃO E DEFESA DO TRABALHO DE CONCLUSÃO DE CURSO II

Aos nove dias do mês de fevereiro do ano de dois mil e vinte e dois, às 16:00 horas, foi realizada a sessão pública de apresentação e defesa do Trabalho de Conclusão de Curso II do Graduando **João Victor de Oliveira Aguiar** (matrícula 20161070080027) do curso de Engenharia Civil, no segundo semestre letivo do ano de dois mil e vinte um. A defesa ocorreu via web Conferência pelo Google Meet. A banca foi composta pelos seguintes membros:

- Vitor Amadeu da Silva Feitosa (Orientador)
- Agno Alves Vieira (Avaliador Interno)
- Waldeyr Mendes Cordeiro da Silva (Avaliador Externo)

Sob a presidência do primeiro. O trabalho de conclusão de curso tem como título "**Automação de análises de projeto no Autodesk Revit**", sob orientação do Prof. Vitor Amadeu da Silva Feitosa. Após a apresentação do Trabalho de Conclusão de Curso II, tendo sido o autor arguido pela Banca Examinadora, a nota obtida foi **9,5**. A sessão foi encerrada às 17: horas. Eu, Prof. André Augusto Nóbrega Dantas, dato e assino a presente ata que segue assinada por todos os membros da Banca.

Prof. André Augusto Nóbrega Dantas

(Assinado eletronicamente)

Prof. Vitor Amadeu da Silva Feitosa

(Assinado eletronicamente)

Prof. Agno Alves Vieira

(Assinado eletronicamente)

Prof. Waldeyr Mendes Cordeiro da Silva

(Assinado eletronicamente)

Prof. Fabiano Campos Macêdo

Coordenador do Curso Bacharelado em Engenharia Civil/ Câmpus Formosa
(Assinado eletronicamente)

Documento assinado eletronicamente por:

- Vitor Amadeu da Silva Feitoza, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 17/02/2022 11:13:25.
- Waldeyr Mendes Cordeiro da Silva, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 15/02/2022 15:47:39.
- Fabiano Campos Macedo, COORDENADOR - FUC1 - FOR-2CCSCI, em 15/02/2022 11:11:33.
- Agno Alves Vieira, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 14/02/2022 10:29:02.
- Andre Augusto Nobrega Dantas, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 13/02/2022 21:54:09.

Este documento foi emitido pelo SUAP em 13/02/2022. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifg.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código Verificador: 246010

Código de Autenticação: 85dea419f2



Instituto Federal de Educação, Ciência e Tecnologia de Goiás

Rua 64 esquina com Rua 11, S/Nº, Expansão Parque Lago, Formosa / GO, CEP 73813-816
(61) 3642-9491 (ramal: 9491), (61) 3642-9493 (ramal: 9493)

JOÃO VICTOR DE OLIVEIRA AGUIAR

AUTOMAÇÃO DE ANÁLISES DE PROJETO NO AUTODESK REVIT

Trabalho de Conclusão de Curso submetido à Coordenação do Curso de Bacharelado em Engenharia Civil do Instituto Federal de Educação, Ciência e Tecnologia de Goiás, Campus Formosa, como parte dos requisitos para a obtenção do grau de Bacharel em Engenharia Civil.

Aprovado em ____ de ____ de ____.

BANCA EXAMINADORA

Prof. Me. Vitor Amadeu da Silva Feitoza. (IFG – Campus Formosa)
Orientador

Prof. Dr. Agno Alves Vieira. (IFG – Campus Formosa)
Examinador

Prof. Dr. Waldeyr Mendes Cordeiro da Silva. (IFG – Campus Formosa)
Examinador

RESUMO

O mundo encontra-se em um momento em que novas tecnologias e inovações surgem cada vez mais rápido. Termos como automação e inteligência artificial passam a ser parte do cotidiano de profissionais de todos os setores da economia. Para profissionais do setor da construção, programas que utilizam a metodologia *BIM* (*Building Information Modeling*) representam mudanças dos paradigmas dos processos tradicionais de criação de projetos. É nesse contexto de constante inovação e adaptação a ferramentas digitais, que habilidades relacionadas à programação de computadores se mostram cada vez mais indispensáveis. Assim, este trabalho é elaborado com o foco de explorar a ideia de modelagem paramétrica, trazida pela metodologia *BIM*, aliada ao poder criativo de uma linguagem de programação. Para tanto, este trabalho demonstra como estender as funcionalidades do Autodesk Revit utilizando sua *Application Programming Interface* (*API*) associada a uma interface de programação na linguagem Python. O objetivo aqui, é automatizar análises de um modelo 3D de uma edificação de acordo com as diretrizes da Resolução N°. 09, de 08 de Abril de 2011, da ADASA, que trata do lançamento de águas pluviais em corpos hídricos do Distrito Federal. Por fim, este trabalho foi capaz de demonstrar os benefícios provenientes da junção de conhecimentos técnicos do setor da construção com a programação de computadores, permitindo maior agilidade e ganho de tempo com a automatização das diretrizes e dos cálculos de um documento técnico.

Palavras-chave: BIM, Automação, Programação, Python, Autodesk Revit, API, Resolução N°. 09 de 08 de Abril de 2011 da ADASA.

ABSTRACT

The world's new technologies and innovations are emerging faster and faster. Terms such as automation and artificial intelligence become part of professionals' daily lives from all sectors of the economy. For professionals in the construction sector, software that uses the BIM (Building Information Modeling) methodology represent paradigm shifts from traditional project creation processes. In this context of constant innovation and adaptation to digital tools, skills related to computer programming are increasingly indispensable. Thus, this work focuses on exploring the idea of parametric modelling, brought by the BIM methodology, combined with the creative power of a programming language. Therefore, it demonstrates how to extend Autodesk Revit functionality using its Application Programming Interface (API) associated with a programming interface in the Python. The objective here is to automate analyzes of a 3D model of a building according to the guidelines of ADASA's Resolution Number 09 of April 08, 2011, which deals with the release of rainwater into water bodies in the Federal District. Finally, this work was able to demonstrate the benefits arising from the combination of technical knowledge of the construction sector with computer programming, allowing greater agility and time saving with the automation of guidelines and calculations of a technical document.

Keywords: BIM, Automation, Programming, Python, Autodesk Revit, API, ADASA's Resolution Number 09 of April 08 2011.

Lista de Figuras

Figura 1 – Esquema de colaboração entre os diversos profissionais envolvidos no processo de construção de uma edificação.	34
Figura 2 – Esquema IPD tornando a metodologia BIM mais centralizada e integrando processos.	35
Figura 3 – Curva de Macleamy.	36
Figura 4 – Modelo 3D de partes das instalações de ar-condicionado de uma edificação, gerado por um software BIM.....	38
Figura 5 – Planilhas e vistas geradas por um software BIM.	38
Figura 6 – Localização e classificação automática de interferências geométricas, geradas por um software BIM.	39
Figura 7 – Dimensões BIM no ciclo de vida de uma edificação.	41
Figura 8 – Modelo BIM 4D: associação do modelo 3D com o cronograma e tarefas de planejamento.....	42
Figura 9 – Lista de Softwares BIM.	44
Figura 10 – Pirâmide com os três conceitos básicos do openBIM.	47
Figura 11 – Cronologia do padrão IFC.	48
Figura 12 – Após importação do modelo IFC, elementos contínuos foram subdivididos em vários elementos e foram criados elementos de intersecção.	49
Figura 13 – Projeto sem acabamento. Modelo IFC visualizado pelo software BIMer.	50
Figura 14 – Grid Radial criado por meio de API no software Tekla Structures.	52
Figura 15 – Gerenciador de tarefas criado por meio de API no software Tekla Structures.	53

Figura 16 – Progressão das revoluções industriais.	61
Figura 17 – Volume de produção nos tipos de sistemas de automação industrial.	63
Figura 18 – Drone.	64
Figura 19 – Máquina de reboco de paredes automática.	65
Figura 20 – Diferenças entre os tipos de linguagens de programação.	67
Figura 21 – Comparação do código escrito em Python com C++ e Java.	71
Figura 22 – Interface do Revit Python Shell.	75
Figura 23 – Exemplo de script que determina o volume das paredes do modelo.	76
Figura 24 – Aba pyRevit com novos comandos, disponibilizada após instalação do plugin.	77
Figura 25 – Rotina elaborada em Python por Sena (2019).	78
Figura 26 – Resultado da execução da rotina criada por Sena (2019).	79
Figura 27 – Fluxograma com as fases deste trabalho.	80
Figura 28 – Exemplo de uma página do documento de referência da API do Revit.	84
Figura 29 – Trecho de código de um dos programas incluídos na instalação do pyRevit.	88
Figura 30 – Exemplo do uso do <i>RevitLookup</i> em um elemento de um modelo Revit.	89
Figura 31 – Exemplo do uso do Revit Python Shell para extrair dados de um elemento.	90
Figura 32 – Exemplo de uma simples interface gráfica para o usuário.	91
Figura 33 – Botão de acesso ao programa desenvolvido, presente na aba “ADASA” ...	93
Figura 34 – Interface do programa desenvolvido.	94
Figura 35 – Pontos importantes da interface gráfica da aplicação.	95

Figura 36 – Interface da janela de resultados.	97
Figura 37 – Mensagem de erro para valores incompatíveis de área do terreno.	98
Figura 38 – Mensagem de erro para terrenos acima de 200 ha.	99
Figura 39 – Vista 3D do modelo cedido para testes.....	100
Figura 40 – Planta baixa do modelo cedido.	101
Figura 41 – Parte da área destaca em azul está sobreposta pela laje da edificação. ...	104
Figura 42 – Área da Figura 41 com a vista da cobertura.....	105
Figura 43 – Levantamento das áreas impermeáveis do modelo de testes.	106
Figura 44 – Resultado da execução do programa no modelo de testes.....	107
Figura 45 – Padrão de arquivo que pode ser carregado no programa desenvolvido...	108
Figura 46 – Arquivo CSV com os dados do trabalho de Braga (2018).	109
Figura 47 –Interface do programa após carregar os dados de Braga (2018).	110
Figura 48 – Resultado da execução do programa com os dados de Braga (2018).....	111

Lista de abreviações

2D – Duas dimensões

3D – Três dimensões

4D – Quatro dimensões

5D – Cinco dimensões

6D – Seis dimensões

7D – Sete dimensões

ADASA – Agência Reguladora de Águas, Energia e Saneamento do Distrito Federal

AEC – Arquitetura, Engenharia e Construção

AIA – American Institute of Architects

API – Application Programming Interface

BIM – Building Information Modeling

CAD – Computer Aided Design

CBIC – Câmara Brasileira da Indústria da Construção

CIS/2 – CIMSteel Integration Standards

CLP – Controlador Lógico Programável

CSV – Comma Separated Values

CWI – National Research Institute for Mathematics and Computer Science in Amsterdam

IAI – International Alliance for Interoperability

IFC – Industry Foundation Classes

IPD – Integrated Project Delivery

ISO – International Organization for Standardization

JSON – JavaScript Object Notation

ML – Machine Learning

REST – Representative State Transfer

RPS – Revit Python Shell

SDK – Software Development Kit

WPF – Windows Presentation Foundation

XAML – Extensible Application Markup Language

Sumário

Capítulo 1 – Introdução.....	19
1.1. Justificativa	21
1.2. Objetivos.....	22
1.2.1. Objetivo geral.....	22
1.2.2. Objetivos específicos	23
Capítulo 2 – Revisão Bibliográfica.....	24
2.1. Resolução Nº 09 de 08 de Abril de 2011, ADASA.....	24
2.1.1. Objetivos e definições gerais	24
2.1.2. Procedimento de cálculo dos reservatórios.....	24
2.1.3. Vazão a ser considerada	26
2.1.4. Redução da área impermeável	28
2.1.5. Mecanização digital das diretrizes de documentos técnicos.....	29
2.2. O contexto da construção civil no Brasil	29
2.3. Metodologia <i>BIM</i>	30
2.3.1. História do <i>Building Information Modeling</i>	31
2.3.2. Fundamentos.....	32
2.3.3. Modelagem Paramétrica de Objetos.....	36
2.3.4. As dimensões de um modelo <i>BIM</i>	40
2.3.5. Interoperabilidade	43
2.3.6. <i>Industry Foundation Classes (IFC)</i>	47
2.3.7. Integração por meio de <i>Application Programming Interface (API)</i>	51
2.3.8. Benefícios da utilização da metodologia <i>BIM</i>	53
2.3.9. Obstáculos relacionados à adoção do <i>BIM</i>	57

2.4. Programação e Automação	58
2.4.1. Revoluções Industriais.....	58
2.4.2. Automação	61
2.4.3. Programação de computadores	66
2.4.4. Linguagem Python	69
2.5. Integração Revit – Python.....	72
2.5.1. Autodesk Revit e sua <i>API</i>	73
2.5.2. <i>Revit Python Shell (RPS)</i>	74
2.5.3. <i>pyRevit</i>	76
Capítulo 3 – Metodologia	80
3.1. Primeira fase	81
3.2. Segunda Fase	81
3.3. Terceira Fase	82
Capítulo 4 – Análises e Resultados	83
4.1. Desenvolvimento de aplicações para o Autodesk Revit	83
4.2. Processo de desenvolvimento do programa proposto	85
4.2.1. Python na utilização da <i>API</i> do Revit.....	86
4.2.2. Criação da interface gráfica do usuário	90
4.2.3. Implementação dos cálculos referentes à Resolução Nº 09	92
4.3. Utilização do programa desenvolvido.....	93
4.4. Viabilidade e limitações	97
4.5. Testes e resultados	99
4.5.1. Primeiro teste: modelo 3D cedido por um arquiteto.....	99
4.5.2. Segundo teste: dimensionamento realizado por Braga (2018)	107
Capítulo 5 – Conclusões	112

Referências..... 115

Capítulo 1 – Introdução

Atualmente, a maioria dos processos que envolvem a concepção de uma edificação, desde a fase projetual até a executiva, são regidos por diretrizes presentes em normas e documentos técnicos. A Agência Reguladora de Águas, Energia e Saneamento do Distrito Federal (ADASA) publicou em 08 de abril de 2011 a Resolução Nº 09, um documento que trata dos requerimentos e da obtenção de outorga para lançamento de águas pluviais em corpos hídricos superficiais e visa o controle quantitativo e qualitativo dos recursos hídricos no Distrito Federal.

Por esse tipo de documento estabelecer diretrizes e critérios de uma forma clara e objetiva, fica evidente que a possibilidade de automatização desses cálculos pode evitar que profissionais tenham a necessidade de repetir a aplicação manual dessas diretrizes para novos projetos. Essas automatizações estão se tornando cada vez mais fáceis devido ao contexto de constante evolução tecnológica e presença de ferramentas digitais que o mundo se encontra hoje.

A quarta revolução industrial está sendo responsável por trazer cada vez mais novas tecnologias e inovações para todos os setores da economia. Sua característica mais marcante é sofisticação e integração de redes e computadores, tornando os processos cada vez mais automatizados devido à forma como os dados são tratados e manipulados (SCHWAB, 2016). Para o setor da construção, o representante dessa forma de atuar, focada em gerir dados de uma forma inteligente, é a metodologia *Building Information Modeling (BIM)*.

Segundo Eastman (2014), um dos mais importantes disseminadores da metodologia *BIM* no mundo, o modelo atual de construção é constituído por um processo fragmentado, onde ocorrem atrasos e custos imprevistos devidos a erros e omissões. Em contrapartida, a implementação da metodologia *BIM* traz um processo em que os dados são dispostos de forma centralizada em um único modelo que contempla todo o ciclo de vida de uma construção, fomentando a colaboração e integração de todos os profissionais envolvidos (EASTMAN et al., 2014).

A aplicação prática da metodologia *BIM* se dá através de *softwares*. Há diversos tipos de programas desenvolvidos por empresas diferentes, que aplicam a metodologia *BIM*. Essa diversidade de *softwares* acontece devido ao foco em diferentes atividades ou processos construtivos. Há sistemas que facilitam a modelagem arquitetônica, outros são focados no cálculo estrutural e assim por diante. O que de fato diferenciam esses *softwares* dos sistemas *CAD* (*Computer Aided Design*) são: a modelagem paramétrica e a interoperabilidade (BAIA, 2015).

Conforme Baia (2015), a modelagem paramétrica é a capacidade de associar regras e parâmetros à geometria dos objetos modelados, tornando-os bem mais do que uma simples representação visual. Com essa tecnologia é possível extrair relatórios, conferir inconsistências entre objetos e fazer automatização de tarefas como cálculos ou levantamentos quantitativos. Já a interoperabilidade é a habilidade de os diferentes *softwares* comunicar e trocar as informações modeladas de forma simultânea, tornando o desenvolvimento integrado e trazendo uma abordagem mais colaborativa. (BAIA, 2015)

Dada a natureza de *softwares BIM* de se basearem em dados e parâmetros para realizar suas modelagens, aliar interfaces de programação com o objetivo de alterar, extrair, acrescentar ou até mesmo automatizar parâmetros de projeto, torna possível a customização de um programa *BIM* que, por sua vez, tem a capacidade de alcançar patamares de qualidade projetual que não seriam possíveis sem a junção da programação com um *software BIM* (LABIB, 2017).

Ainda que os benefícios e o potencial da utilização de linguagens de programação associada a *softwares BIM* sejam claros, os profissionais enfrentam um grande obstáculo: o conhecimento técnico ligado ao uso da programação. O uso de linguagens de programação requer o entendimento de vários conceitos dessa área e automatizar tarefas pode ser um processo complexo. Os projetistas que optam por utilizar linguagens de programação, encontram o desafio de aliar suas competências profissionais à programação de computadores (SENA, 2019).

Uma das formas de associar uma interface de programação a um *software*, é por meio das chamadas *APIs* (*Application Programming Interfaces*), que permitem a troca de

informações ao expor um conjunto de dados e funções, com o objetivo de auxiliar na interação entre programas computacionais (MASSÉ, 2011). Essa tecnologia é o núcleo do desenvolvimento moderno de *softwares*, permitindo a interoperabilidade entre plataformas e integração de serviços de terceiros em uma aplicação (JIN et al., 2018).

De acordo com Catelani (2016), diversas soluções que tentam aplicar a metodologia *BIM* possuem *APIs* que podem ser utilizadas, permitindo a integração entre sistemas, fomentando a interoperabilidade e estendendo as funcionalidades de um *software*, com recursos que não eram originalmente utilizáveis. Assim, o uso de *APIs*, por meio da programação de computadores, tem um grande potencial em auxiliar e preparar os profissionais para os desafios provenientes do uso de ferramentas digitais (LABIB, 2017).

1.1. Justificativa

A metodologia *BIM* não é algo recente, em diversos países já é obrigatória a sua aplicação para obras públicas há anos e o Brasil está se encaminhando para essa mesma realidade. Em 22 de agosto de 2019, foi publicado o decreto nº 9.983 que dispõe sobre a Estratégia Nacional de Disseminação do BIM. Adicionalmente, em 02 de abril de 2020, o Diário Oficial da União publicou o decreto nº 10.306 que estabelece a utilização do *BIM* na execução direta ou indireta de obras e serviços de engenharia realizada pelos órgãos e pelas entidades de administração pública federal.

Schwab (2016) afirma que a atual revolução industrial que o mundo está vivendo, apresenta o surgimento e difusão de novas tecnologias de forma extremamente rápida, tendo seus efeitos manifestados em forma de automações e criações de processos sem precedentes. As soluções baseadas em *BIM* também seguem esse padrão, surgindo cada vez mais programas para os diversos setores da construção, sendo iminente a migração dos *softwares* tradicionais para os que se baseiam na metodologia *BIM*, já que, mesmo sem ter seu potencial explorado por completo, a metodologia já se provou eficaz (DIAS, 2020).

É nesse contexto de grande competitividade e surgimento de novas tecnologias, onde tudo é baseado no uso e na integração de computadores, que se observa a necessidade de conhecimentos na área de programação para que se obtenha o domínio dessas ferramentas e seja possível realizar um uso inteligente e otimizado delas. Para Labib (2017), no campo da arquitetura e construção civil, essas habilidades relacionadas à programação são cruciais para tornar os processos de modelagem mais eficientes, além de facilitar os processos de organização, análise e visualização.

Portanto, ainda que existem diversas barreiras para a implementação de soluções *BIM*, principalmente no Brasil onde há a falta de formação de profissionais habilitados para utilizar as ferramentas (CARNEIRO; MACIEL, 2020), o desenvolvimento de pesquisas e a disseminação de conhecimentos em como utilizar as ferramentas e, ainda, em como estender suas funcionalidades por meio de interfaces de programação, se tornam fatores essenciais para atingir o verdadeiro potencial da metodologia *BIM* e levar o setor da construção civil para outros patamares em termos de precisão, integração, colaboração, velocidade, eficácia e redução de custos.

Além disso, a possibilidade de automatizar, em um *software* baseado na metodologia *BIM*, os cálculos e as verificações presentes em documentos técnicos como a Resolução Nº 09 de 08 de abril de 2011 da ADASA, gera a oportunidade de criar produtos digitais voltados para profissionais do setor da construção que têm o potencial de tornar o processo de aplicação de normas mais rápido, dinâmico e preciso.

1.2. Objetivos

1.2.1. Objetivo geral

Criar um produto para o Autodesk Revit que seja capaz de aplicar e automatizar as diretrizes e os cálculos da Resolução Nº 09 de 08 de abril de 2011, da ADASA que trata do lançamento de águas pluviais em corpos hídricos do Distrito Federal. Com isso, proporcionar aos profissionais do setor da construção uma ferramenta que trará maior agilidade quanto a adequação dessa resolução em seus projetos.

1.2.2. Objetivos específicos

- Utilizar a linguagem de programação Python para realizar a interação com a *API*.
- Adotar o *plugin Revit Python Shell (RPS)* com a finalidade de realizar testes e estudos quanto as funcionalidades da *API* do Revit.
- Adotar o *plugin pyRevit* para criar um botão no *software* que executará um código já escrito em Python, diretamente no Revit.
- Criar um botão, por meio do *pyRevit*, que executará um código em Python para aplicar as diretrizes e realizar os cálculos referentes à Resolução N°. 09, de 08 de Abril de 2011, estabelecida pela ADASA, no modelo 3D presente no *software* Revit.
- Desenvolver uma interface gráfica para auxiliar o usuário na utilização do programa.
- Verificar os resultados obtidos e avaliar o potencial do uso da *API* disponibilizada pelo Autodesk Revit.

Capítulo 2 – Revisão Bibliográfica

2.1. Resolução N° 09 de 08 de Abril de 2011, ADASA

2.1.1. Objetivos e definições gerais

A resolução da ADASA tem como foco o lançamento de águas pluviais em corpos hídricos do domínio do Distrito Federal, estabelecendo procedimentos gerais para o requerimento e obtenção de outorga para essa atividade, tendo em vista o controle qualitativo e quantitativo dos recursos hídricos do Distrito Federal.

O documento, em seu artigo 2º, traz diversas definições, das quais vale a pena destacar as seguintes:

V - reservatório de qualidade: reservatório que retenha determinado volume de água originado pelo escoamento superficial proveniente de impermeabilização do solo, com o objetivo principal de reduzir a carga poluente a ser lançada no corpo hídrico receptor;

VI - reservatório de quantidade: reservatório que tem como objetivo principal reter determinado volume de água originado pelo escoamento superficial, reduzindo as vazões de pico e retardando o escoamento das águas pluviais provenientes de impermeabilização do solo, de forma a amenizar possíveis impactos no corpo hídrico receptor;

VIII - vazão de pré-desenvolvimento: vazão estimada de escoamento superficial calculada considerando-se a situação natural de cobertura do solo;

2.1.2. Procedimento de cálculo dos reservatórios

Os artigos 7º e 8º da Resolução N° 09 tratam dos reservatórios de qualidade e de quantidade, respectivamente. O volume do reservatório de qualidade e a vazão de saída máxima do mesmo para um reservatório de quantidade são descritos, respectivamente, de acordo com as seguintes equações:

$$V_{qa} = (33,8 + 1,80 \times A_i) \times A_c \quad (1)$$

$$Q = \frac{V_{qa}}{86,4} \quad (2)$$

Onde:

V_{qa}: volume em m³ (metro cúbico).

A_i: percentual de área impermeável do terreno.

A_c: é a área de contribuição do empreendimento em ha (hectare).

Q: vazão em L/s (litros por segundo).

Já para o dimensionamento do reservatório de quantidade, o artigo 5º também deverá ser observado. Assim, o lançamento no corpo hídrico não pode ultrapassar a vazão máxima específica de 24,4 L/(s.ha) (vinte e quatro inteiros e quatro décimos de litro por segundo por hectare). Para áreas de contribuição inferiores a 200 ha, volume do reservatório de quantidade será de acordo com a seguinte equação:

$$V = (4,705 \times A_i) \times A_c \quad (3)$$

Onde:

V: volume em m³ (metro cúbico).

A_i: percentual de área impermeável do terreno.

A_c: é a área de contribuição do empreendimento em ha (hectare).

Para áreas superiores a 200 ha, é necessária a elaboração de um estudo hidrológico para determinar o volume do reservatório de quantidade e seus dispositivos de saída, visando garantir que a vazão a ser lançada no corpo hídrico não ultrapasse a vazão de pré-desenvolvimento.

2.1.3. Vazão a ser considerada

De acordo com o artigo 5º da Resolução Nº 09, o limite para a vazão específica a ser lançada é de até 24,4 L/(s.ha) (litro por segundo por hectare). Também é mencionado nesse mesmo artigo que para terrenos com área inferior a 600 m², esse limite pode ser desconsiderado, a critério da ADASA.

Um dos métodos de estimativa do escoamento superficial é o método racional, descrito por Tomaz (2011). Segundo Tomaz (2011), é um método indireto que estabelece uma relação entre a chuva e o escoamento superficial, sendo utilizado para calcular a vazão de pico de uma determinada bacia. Esse método deve ser aplicado em bacias pequenas com área de drenagem inferior a 300 ha ou com tempo de concentração inferior a uma hora (TOMAZ, 2011). A equação que descreve esse método, de acordo com Tomaz (2011), é a seguinte:

$$Q = \frac{C \times I \times A}{3600} \quad (4)$$

Onde:

Q: vazão em L/s (litros por segundo).

C: coeficiente de escoamento superficial.

I: intensidade média da chuva em mm/h (milímetro por hora).

A: área da bacia em m² (metro quadrado).

Em relação ao coeficiente de escoamento superficial, o Termo de Referência e Especificações para Elaboração de Projetos de Sistema de Drenagem Pluvial no Distrito Federal, publicado em 2019 pela NOVACAP, traz valores que podem ser usados como referência. Esses valores podem ser observados a seguir:

Tabela 1 – Coeficientes de escoamento superficial.

0,9	áreas calçadas ou impermeabilizadas
0,78	áreas com bloco intertravado maciço
0,70	áreas urbanizadas com áreas verdes
0,40	áreas com bloco intertravado vazado com preenchimento de areia ou grama
0,30	áreas de solo natural com recobrimento de brita
0,20	áreas com inclinação superior a 5% integralmente gramadas ou com jardins ou vegetação natural
0,15	áreas com inclinação inferior a 5% integralmente gramadas ou com jardins ou vegetação natural

Fonte: NOVACAP, 2019.

Já em relação a intensidade da chuva, o termo mencionado anteriormente também traz uma equação que descreve esse dado. Todavia, a unidade descrita está em L/s.ha, para se obter a intensidade da chuva em mm/h mais facilmente é possível consultar a versão desse mesmo Termo de Referência publicada em 2017. Assim, a equação que descreve a intensidade da chuva se dá da seguinte forma:

$$i = \frac{21,7 \times F^{0,16}}{(tc + 11)^{0,815}} \quad (5)$$

Onde:

i: intensidade da chuva em mm/min (milímetro por minuto).

F: período de retorno em anos.

tc: tempo de concentração em minutos.

Por fim, o artigo 4º da Resolução Nº 09 informa que se deve considerar chuvas com tempo de recorrência de 10 (dez) anos. Em relação ao tempo de concentração, a NBR 10844:1989 informa que deve ser fixado em 5 (cinco) minutos.

2.1.4. Redução da área impermeável

Outro ponto importante da Resolução Nº 09 é em relação às áreas impermeáveis. De acordo com o artigo 8º da Resolução Nº 09, referente ao cálculo dos reservatórios de quantidade, a área impermeável deve ser estabelecida de acordo com o somatório das áreas impermeáveis previstas no projeto. Por fim, o capítulo 5 afirma que o percentual de área impermeável a ser considerado no cálculo pode ser reduzido, caso sejam implementadas medidas que favoreçam a infiltração de água no solo. De acordo com a resolução, essas medidas podem ser:

- I – aplicação de pavimentos permeáveis (blocos vazados com preenchimento de areia ou grama, asfalto poroso, concreto poroso) – reduzir em até 60% (sessenta por cento) a área que utiliza estes pavimentos;
- II – desconexão das calhas de telhado de forma a direcionar a água para superfícies permeáveis com drenagem – reduzir em até 40% (quarenta por cento) a área de telhado drenada;
- III – desconexão das calhas de telhado de forma a direcionar a água para superfícies permeáveis sem drenagem – reduzir em até 80% (oitenta por cento) a área de telhado drenada;
- IV – aplicação de trincheiras de infiltração – reduzir em até 80% (oitenta por cento) as áreas drenadas para as trincheiras;
- V – direcionamento da água proveniente de superfície impermeável para dispositivos de infiltração sem saída – percentual de redução a ser estimado pela ADASA;
- VI – aplicação de outras medidas a serem avaliadas pela ADASA.

2.1.5. Mecanização digital das diretrizes de documentos técnicos

O setor da construção civil tem sua complexidade fundamentada na grande quantidade de subsistemas e disciplinas envolvidas no processo de constituição de uma edificação. Para a maioria dessas disciplinas, seja cálculo estrutural, instalações elétricas, instalações hidráulicas etc., existem diversos documentos técnicos que trazem regras acerca de como esses processos devem ser executados tendo em vista qualidade e segurança.

Devido ao caráter lógico que esses documentos apresentam, a exemplo da resolução tratada neste tópico, trazendo regras, condições, valores a serem comparados e equações a serem seguidas, a utilização de ferramentas digitais para automatizar todo o processo de levantamento de dados e realização de cálculos referentes às diretrizes desses documentos se torna um aparato que permite profissionais do setor da construção realizarem a adequação de seus projetos de forma mais rápida, precisa e automática. Ainda assim, o setor apresenta certa resistência quanto a adoção dessas técnicas.

2.2. O contexto da construção civil no Brasil

A indústria da construção civil tem um grande impacto econômico no país, tendo em vista que engloba todas as atividades presentes na vida útil de uma obra, desde a elaboração do projeto até operações de manutenção. Segundo a Câmara Brasileira da Indústria da Construção (CBIC), em 2020 a construção civil foi responsável por 3,3% do PIB do Brasil.

Apesar de sua importância, o setor da construção civil no Brasil apresenta como suas características principais atrasos na entrega, baixa produtividade e grandes desperdícios (MENDES, 2016). Esses problemas usualmente têm sua origem em projetos inadequados que, quase sempre, contribuem para o aumento de custos e surgimento de manifestações patológicas (LÜKE; NASCIMENTO, 2012). Adicionalmente, as tomadas de decisões, em geral, não apresentam o nível de detalhe necessário e são feitas sem o apoio de informações consistentes (BIOTTO; FORMOSO; ISATTO, 2012).

O mercado brasileiro apresenta uma grande resistência à adoção de novas técnicas e tecnologias, ficando preso em práticas e processos arcaicos de produção. Por ser um setor muito sensível a situação financeira do país, deve-se ter como foco a exatidão e a minimização de erros e prejuízos (ARAUJO, 2018), deixando evidente a necessidade de modernização, através de métodos que aumentem a eficiência e diminuam tempo e custos.

Um dos problemas mais específicos na indústria de construção civil, é a interpretação incorreta de cronogramas e a dificuldade de visualização do planejamento da obra no espaço, devido ao grande número de atividades e precedências (BRITO; FERREIRA, 2015). Já é conhecida a falta de compromisso, por grande parte do setor, com gestão e planejamento de projetos, sendo este o ponto que exige urgentemente mudanças.

A mudança do foco da parte executiva para a parte de planejamento e gestão, aliada a tecnologias que contribuam para este fim, serão os pilares do futuro da indústria da construção civil no Brasil. Em termos de tecnologia, a metodologia *BIM* é a representante do setor da construção, tendo em seus fundamentos baseados em planejamento, gestão, colaboração e eficiência

2.3. Metodologia *BIM*

A forma como os projetos eram elaborados sofreram uma grande mudança com a introdução de *softwares* que utilizam a tecnologia *CAD* (*Computer Aided Design*). Esse modelo se baseia em plantas de projeto em duas dimensões (2D) e facilitou muito o trabalho dos profissionais das áreas de arquitetura e engenharia, porém apresenta diversos problemas como: conflito entre projetos de instalações e estrutural, orçamento e planejamento fica a cargo de uma equipe que tem que imaginar o modelo 3D para realizar o levantamento de quantitativos, perda de projetos originais devido a uma reforma e outros problemas (BAIA, 2015).

Ainda que a tecnologia *CAD* tenha sido um grande avanço, pode-se dizer que ela apenas efetuou um processo de migração do trabalho manual de desenho de projetos

para os computadores, preservando a essência de uma metodologia antiquada. Os sistemas de computadores já provaram seu potencial em diversos setores, falta para a construção civil a disseminação de tecnologias que usam esses sistemas de forma mais inteligente e eficiente, mudando a forma de se pensar e elaborar obras, desde sua concepção até seu uso final.

A metodologia *BIM* (*Building Information Modeling*) é uma das tecnologias mais promissoras para a indústria da construção civil, representando de fato um avanço e uma evolução em relação a metodologia antiga de trabalho. Com essa tecnologia, é possível criar um modelo virtual preciso de uma edificação, contendo a geometria exata, dados relevantes e necessários para dar suporte para todas as etapas da construção, modelando todo o ciclo de vida de uma obra (EASTMAN et al., 2014).

O processo atual de implementação de uma edificação é fragmentado e utiliza formas de comunicação que tendem a ter erros e omissões, gerando custos adicionais, atrasos e imprevistos. Outro problema associado a isso é o tempo considerável e gasto requerido para gerar informações críticas durante a fase de projeto. Por outro lado, quando o processo baseado na metodologia *BIM* é implementado de forma apropriada, tem a capacidade de gerar projetos e construções mais integradas, resultando em melhor qualidade e custos e prazos reduzidos (EASTMAN et al., 2014).

2.3.1. História do *Building Information Modeling*

A ideia inicial do que viria a ser o *BIM* começou por volta dos anos 70, na Universidade de Carnegie-Mellon em Pittsburgh, a partir de uma pesquisa realizada por Charles M. Eastman, um dos maiores nomes quando se trata de *BIM*. Seus estudos preconizavam, dentre outras coisas, o caráter colaborativo e de integração entre todas as partes constituintes de um edifício (FARIAS, 2020).

Foi no final dos anos 70 que a teoria do *Building Information Modeling* foi desenvolvida por estudiosos do Instituto de Tecnologia da Georgia. A metodologia progrediu rapidamente e logo a indústria da construção percebeu o valor de ter um processo de construção integrado (HARRIS, 2010).

O termo “*Building Information Modeling*” só foi de fato utilizado em 2002, para descrever modelos virtuais de design, construção e gestão (HARRIS, 2010). Foi Phill Bernstein, arquiteto da Autodesk, que usou o termo pela primeira vez, sendo popularizado posteriormente por Jerry Laiserin, se referindo a representações digitais dos processos de construção (BARBOSA, 2014).

Atualmente há uma gama de empresas que criam ferramentas que se baseiam na metodologia *BIM*. Para cada parte de uma edificação existe uma ferramenta que se destaca, desde os projetos arquitetônicos, estruturais e de instalações até para atividades de compatibilização, planejamento e orçamento. Devido a problemas de comunicação entre esses diferentes programas, a organização *buildingSMART* trouxe soluções, ao criar um padrão de arquivos internacional e aberto, possibilitando a troca de informações entre projetos criados em *softwares* de empresas diferentes que aderirem ao padrão (LAAKSO e KIVINIEMI, 2012).

2.3.2. Fundamentos

Devido ao caráter de metodologia, o conceito *Building Information Modeling* está continuamente sendo aprimorado, assim se tornando um tanto quanto difícil obter uma definição objetiva de *BIM*. Vários autores definem a metodologia de sua maneira, em geral, focando em um ponto ou em outro. Neste trabalho qualquer referência à tecnologia, modelo ou software *BIM*, deve ser entendido como baseado na metodologia *BIM*.

O livro “Manual de *BIM*” (EASTMAN et al., 2014) traz a definição utilizada pela empresa americana de construção *M.A Mortenson Company*, que aborda seis características principais:

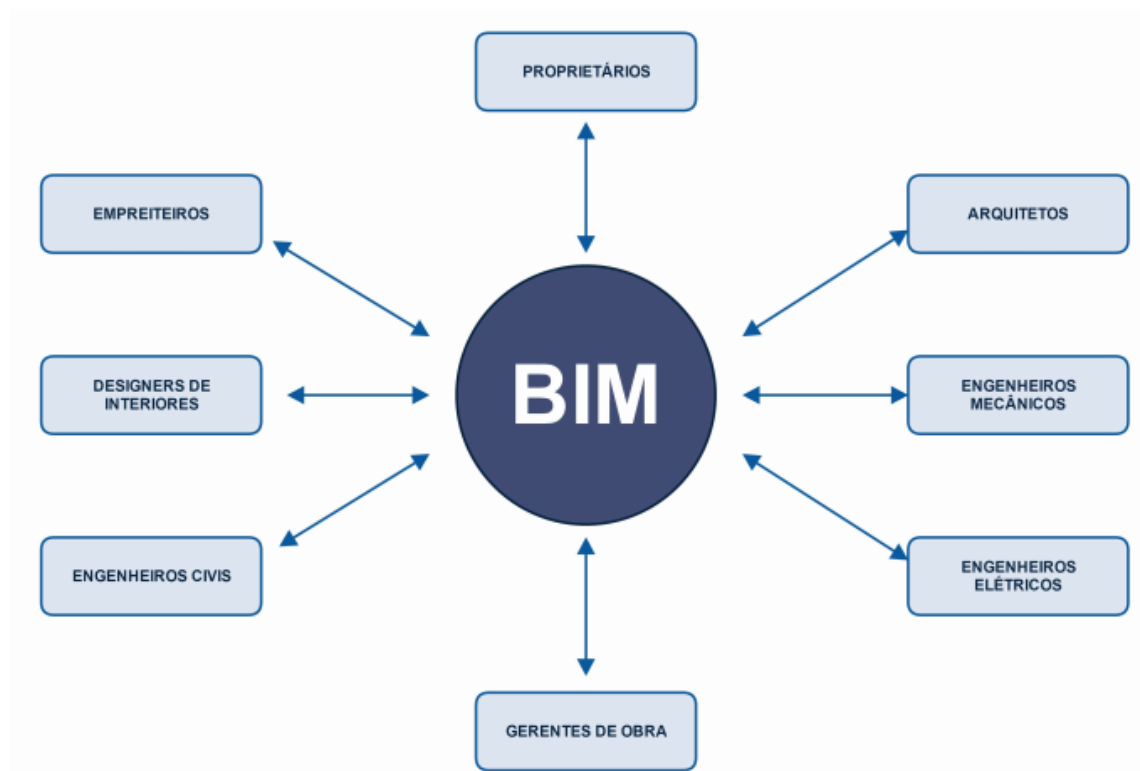
- Digital;
- Espacial (3D);
- Mensurável (quantificável, dimensionável e consultável);

- **Abrangente** (incorporando e comunicando a intenção de projeto, o de desempenho da construção, a construtibilidade, e incluir aspectos sequenciais e financeiros de meios e métodos);
- **Acessível** (a toda a equipe do empreendimento e ao proprietário por meio de uma interface interoperável e intuitiva);
- **Durável** (utilizável ao longo de todas as fases da vida de uma edificação). (EASTMAN et al, 2014)

Não há implementações de *software* que, atualmente, tem a capacidade de abranger todos esses pontos, porém essas características são essenciais para atingir, de forma ideal, um processo construtivo integrado (EASTMAN et al., 2014).

A execução desses critérios de forma efetiva trará uma mudança não só nos *softwares* e suas ferramentas, como também em como todo o ambiente de trabalho é organizado, já que promove um uso cada vez mais digital do modelo, com maior comunicação e colaboração entre os envolvidos (Figura 1), migrando o modelo de projeto para uma forma mais descentralizada. Assim, as etapas de projeto e planejamento se tornarão mais extensas, resultando em projetos mais completos e precisos em suas documentações, orçamentos e quantitativos.

Figura 1 – Esquema de colaboração entre os diversos profissionais envolvidos no processo de construção de uma edificação.



Fonte: Freire (2015).

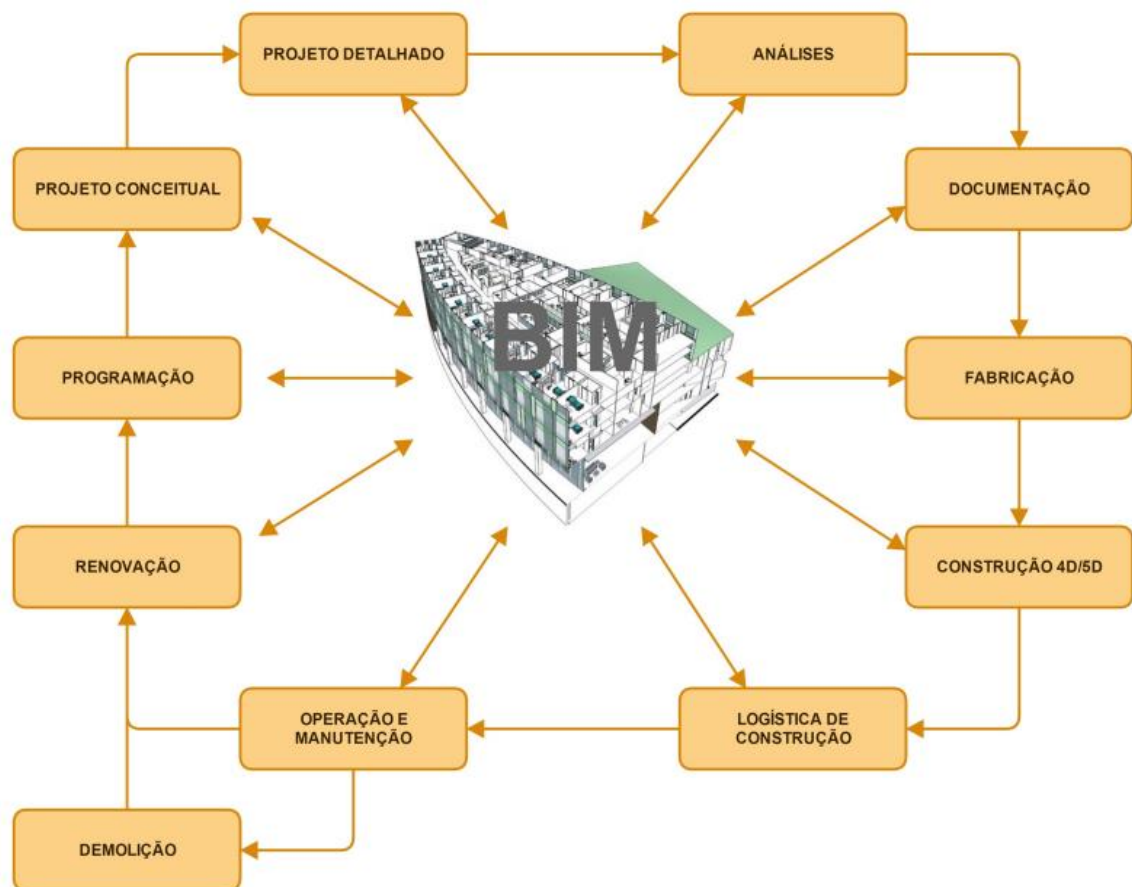
Para Catelani (2016), a metodologia *BIM* pode ser definida de acordo com os seguintes aspectos:

- **Conjunto de políticas, processos e tecnologias:** esses elementos combinados constituem uma metodologia capaz de gerenciar todos os processos e informações de uma edificação, por todo seu ciclo de vida.
- **Processo progressivo:** uma única plataforma que possibilita a modelagem, armazenamento, troca, consolidação e fácil acesso às informações de uma instalação.
- **Nova plataforma de tecnologia da informação aplicada à construção civil:** é materializada em *softwares* que permitem que os processos baseados em documentos sejam realizados de maneiras mais eficazes.

Outro conceito fundamental para aplicação efetiva da metodologia *BIM* é o *Integrated Project Delivery (IPD)*, sendo um método de relação contratual que integra

pessoas, sistemas, estruturas de negócios e práticas, tornando todo o processo de criação e execução de projetos mais centralizado (Figura 2). O *IPD* tem o objetivo de otimizar resultados e reduzir gastos, além de aproximar os envolvidos e fomentar a colaboração entre proprietários, construtores e projetistas (FREIRE, 2015).

Figura 2 – Esquema IPD tornando a metodologia BIM mais centralizada e integrando processos.

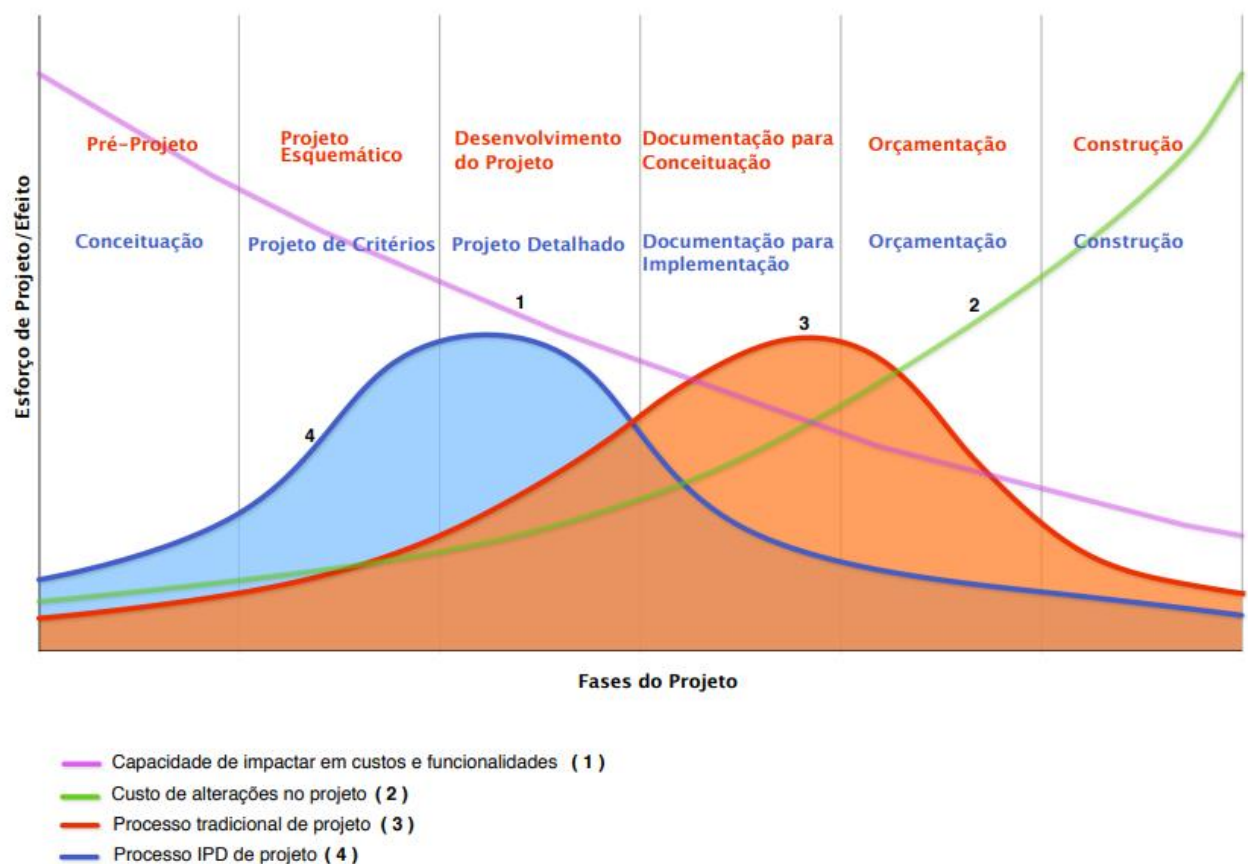


Fonte: Freire (2015).

A aplicação do *IPD* tem consequências no fluxo de trabalho, em relação ao modelo atual, em como o esforço e o gasto de tempo devem ser direcionados ao longo da vida útil de uma edificação, focando nos processos de planejamento e projeto, ao invés de execução (FREIRE, 2015). A curva de Macleamy (Figura 3) consegue demonstrar como

o pico de esforço na fase concepção conduz a custos reduzidos quando comparado com o processo tradicional (RIBEIRO, 2012).

Figura 3 – Curva de Macleamy.



Fonte: Freire (2015).

2.3.3. Modelagem Paramétrica de Objetos

O ponto mais importante para entender a diferença entre a metodologia *BIM* e projetos 2D convencionais é o conceito de modelagem paramétrica. Nesse tipo de modelagem os objetos não possuem propriedades fixas, mas sim regras e parâmetros que determinam suas características, assim os objetos conseguem se relacionar entre si e se atualizarem de forma automática, permitindo uma modelagem mais complexa (MENDES, 2016).

De acordo com Eastman et al. (2014), o método de trabalho em um projeto paramétrico é baseado na criação de uma família de modelos ou classe de elementos, que é uma coleção de regras e relações que controlam os parâmetros das instâncias de um elemento de uma família. As regras podem também ser requisitos de projeto, verificando se o projeto está de acordo com o que foi pré-estabelecido e avisando ao usuário, caso contrário.

Os parâmetros que definem um objeto podem ser relacionados a geometria, distância, ângulos, materiais etc. Já as regras, podem ser algo como: *vinculado a*, *paralelo a* e *distante a* (EASTMAN et al., 2014). Esses tipos de regras configuram as relações entre elementos diferentes, tornando possível a alteração automática de um objeto “a” após a alteração de um objeto “b” ao qual está vinculado.

Outro elemento fundamental para modelos *BIM* é a visualização 3D, que é possível ser gerada de forma automática graças ao caráter paramétrico das aplicações. Assim, qualidade do produto se torna superior com a possibilidade de se observar com mais facilidade e mais rapidamente as mudanças feitas tridimensionalmente, gerando maior produtividade quando comparado aos processos em *CAD* (SENA, 2019). A Figura 4 demonstra a geração de uma vista em 3D por um *software BIM*.

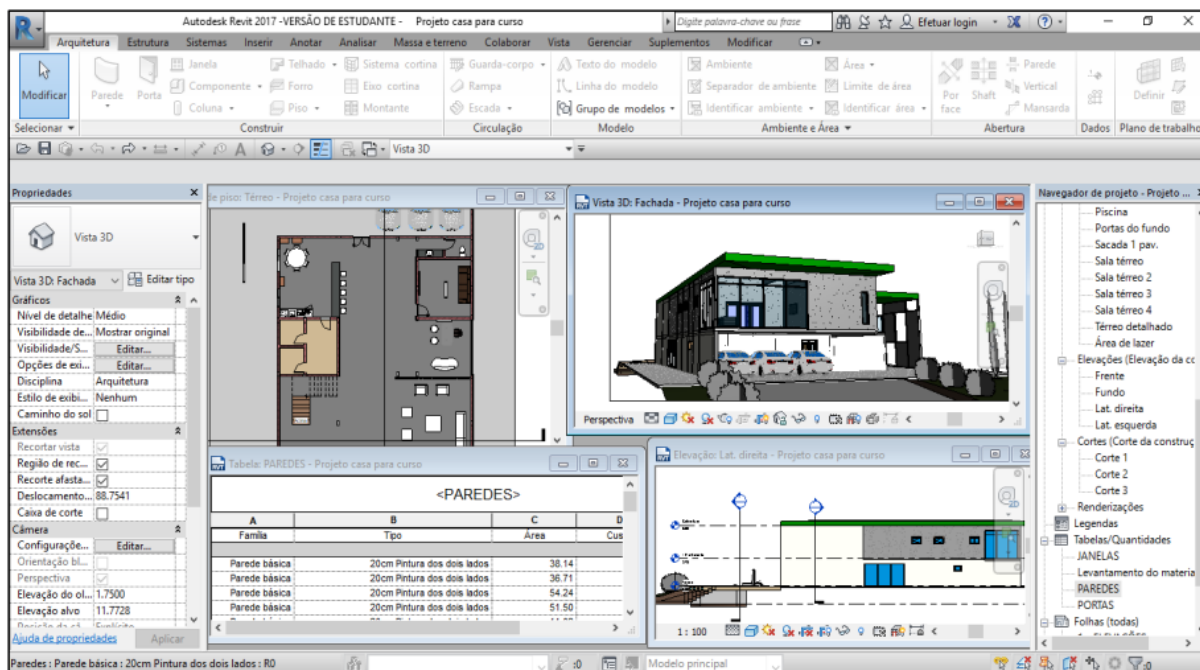
Além da geração automática de uma visualização tridimensional, os *softwares BIM* também possibilitam a geração de vistas laterais e cortes em uma edificação. Como há diversas informações vinculadas aos objetos, também é possível gerar planilhas contendo todos esses dados. A Figura 5 ilustra algumas gerações automáticas que são possíveis com *softwares BIM*.

Figura 4 – Modelo 3D de partes das instalações de ar-condicionado de uma edificação, gerado por um software BIM.



Fonte: Catelani (2016).

Figura 5 – Planilhas e vistas geradas por um software BIM.



Fonte: Sena (2019).

Outra característica importante de sistemas baseados em *BIM* é a possibilidade de, por meio das informações de localização dos objetos no projeto, verificar e detectar conflitos entre os diversos subsistemas de um projeto, contribuindo para rapidez, qualidade e mitigação de erros (SENA, 2019).

Catelani (2016), afirma que relatórios de interferências podem ser extraídos automaticamente de um modelo baseado em *BIM* e alguns *softwares* oferecem formatos padronizados, contendo a imagem do conflito e sua localização, além de classificar as interferências como leves, moderadas ou críticas (Figura 6).

Figura 6 – Localização e classificação automática de interferências geométricas, geradas por um software BIM.



Fonte: Catelani (2016).

Além dos conflitos geométricos (quando um objeto sobrepõe o outro), algumas aplicações *BIM* tem a capacidade de identificar as interferências funcionais, que se configura quando há o impedimento do funcionamento adequado de um objeto. Catelani (2016) traz um exemplo desse tipo de detecção: quando um projetor fixado no teto é posicionado atrás de uma luminária, seu funcionamento estará comprometido.

De uma forma geral, cada objeto em um modelo *BIM* possui regras e parâmetros que gerenciam características geométricas e outras informações, que podem ser compartilhadas com objetos similares. Assim, diferentemente dos modelos *CAD*, aplicações

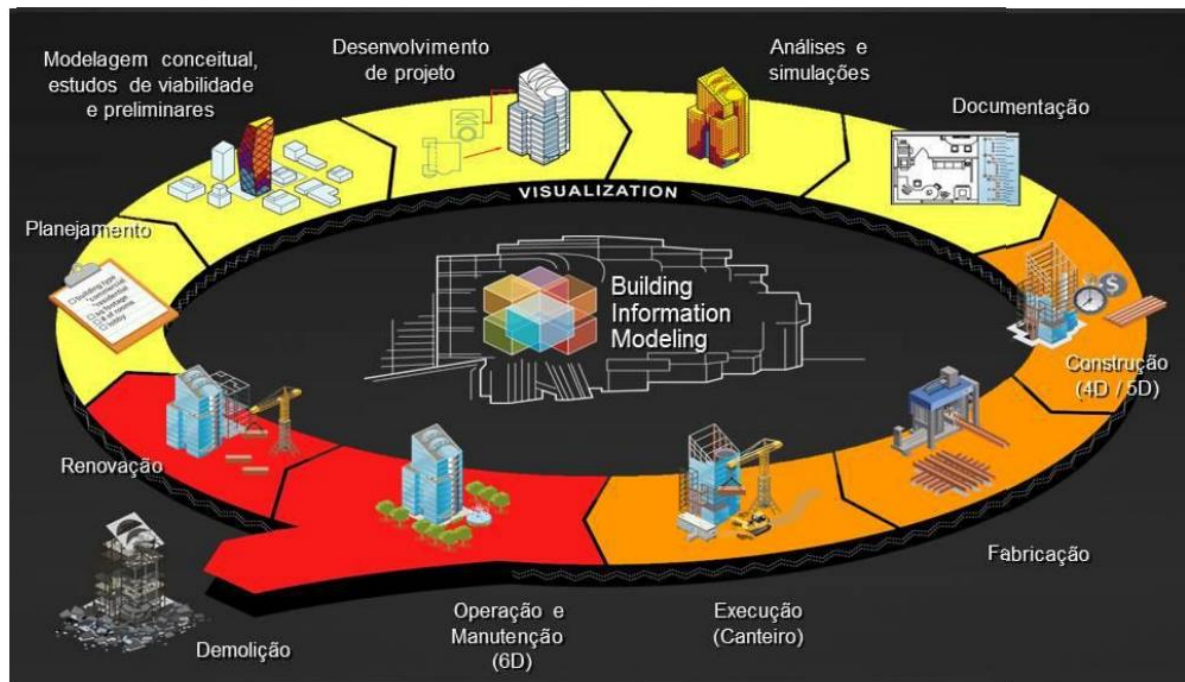
em *BIM* não atuam apenas como uma representação visual de um projeto, mas sim como um banco de dados que incorpora informações sobre as características de objetos e como eles se relacionam entre si.

O entendimento de um modelo *BIM* como banco de dados é fundamental para a aplicação efetiva da metodologia, já que a capacidade de manipular e gerenciar os dados, traz para o usuário possibilidades de automatizar tarefas, gerar relatórios e levantamentos automáticos, tornando o projeto cada vez mais preciso e menos propenso a erros humanos.

2.3.4. As dimensões de um modelo *BIM*

Para Kymmel (2008), o conceito de dimensões em um modelo *BIM* está relacionado a sua disponibilidade e a conexão de informações em um projeto, sendo uma característica de modelos *BIM*. Já para Campestrini et al. (2015), as dimensões se referem a como um modelo está programado e que tipos de informações ele fornece. A Figura 7 ilustra como algumas dimensões de um modelo *BIM* se encaixam em seu ciclo de vida.

Figura 7 – Dimensões BIM no ciclo de vida de uma edificação.

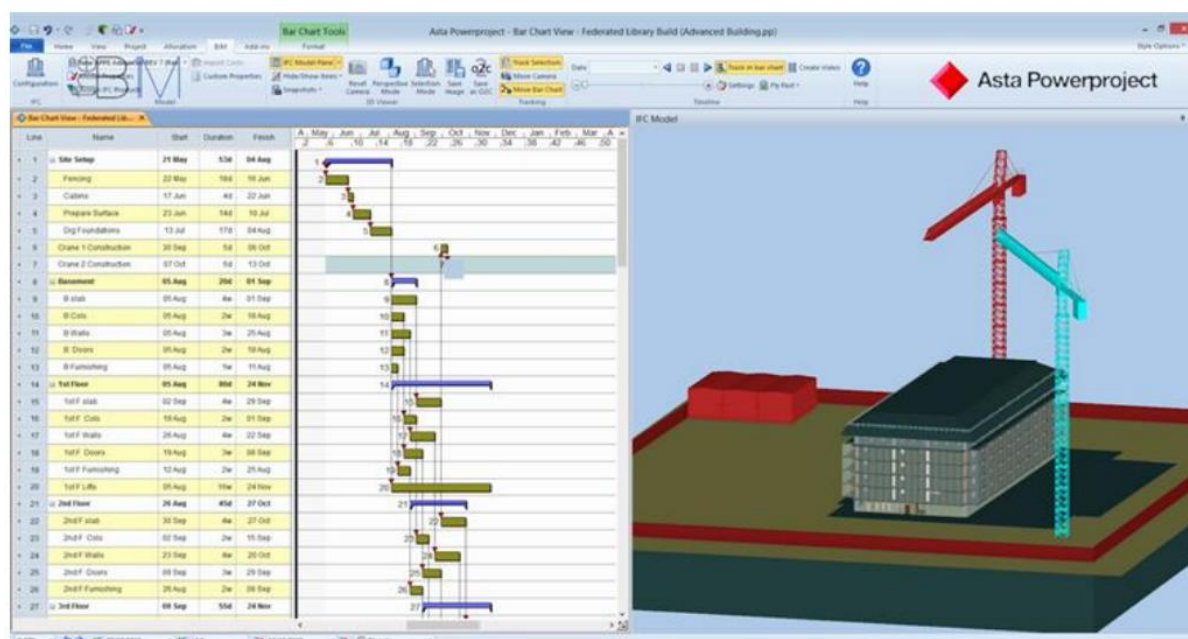


Fonte: Baia (2015).

O *BIM* 3D se relaciona com as informações espaciais e qualidade do projeto (pilares, vigas, lajes, paredes, portas, janelas etc.). Há a possibilidade de extrair informações sobre especificações, materiais, acabamentos, quantitativos, geração de pranchas 2D entre outras informações (CAMPESTRINI et al., 2015). O modelo 3D corresponde desde a modelagem conceitual até a criação da documentação da edificação, também deve permitir a utilização dessas informações para as fases posteriores de um projeto: planejamento, fabricação de componentes, orçamentação e manutenção (BAIA, 2015).

A quarta dimensão (*BIM* 4D) terá como foco o planejamento de uma obra, gerenciando informações relacionadas a prazos, cronogramas e configurações espaciais a cada etapa da execução (CAMPESTRINI et al., 2015). Nessa fase é possível efetuar a simulação do processo de execução, integrando as informações do *BIM* 3D com o cronograma e possibilitando maior eficiência no processo construtivo (BAIA, 2015). A Figura 8 ilustra um exemplo de modelo *BIM* 4D.

Figura 8 – Modelo BIM 4D: associação do modelo 3D com o cronograma e tarefas de planejamento.



Fonte: The B1M (2016) apud Silva (2019).

O *BIM* 5D está associado à adição de custos ao modelo *BIM* 4D, permitindo consultar informações de custo de serviços (materiais, mão de obra, equipamentos etc.) e até mesmo geração de curvas ABC (CAMPESTRINI et al., 2015). Também permite a emissão de relatórios de custos e a visualização do andamento das atividades de construção relacionadas aos custos e ao tempo (BAIA, 2015).

Já o *BIM* 6D está relacionado a informações sobre o uso da edificação. Esse modelo fornece dados sobre custos de operação e manutenção, podendo gerenciar a validade dos materiais, ciclos de manutenção, consumo de água e energia elétrica etc. (CAMPESTRINI et al., 2015).

A sétima dimensão (7D) no modelo *BIM* representa a adição informações sobre análises de impacto ambiental e eficiência energética das edificações, tendo foco em sustentabilidade. Podem ser extraídos dados sobre ventilação, iluminação natural, acústica etc., facilitando a obtenção de certificados de eficiência energética (RIBEIRO, 2015).

Campestrini et al. (2015) afirmam: “Quanto mais dimensões tiver o modelo, maiores serão os tipos de informações possíveis de serem modeladas a partir deles, tornando as tomadas de decisão mais complexas e acertadas.”

2.3.5. Interoperabilidade

A implementação de um *software* que aplique a metodologia *BIM* para a construção exige um trabalho complexo e extenso, dada todas as nuances que envolvem um projeto de uma edificação. Pelo grau de confiabilidade que esses programas devem oferecer, as grandes empresas envolvidas na criação desses *softwares* ficam impossibilitadas de criar uma única aplicação que aborde todos os subsistemas de uma obra, de forma satisfatória. Portanto, atualmente existem diversos programas que apresentam soluções baseadas na metodologia *BIM*, focadas em desenvolver ferramentas para partes mais específicas de um projeto.

De uma forma geral, os programas baseados em *BIM* têm a capacidade de tratar de diversas fases de um projeto e, alguns, tem até mesmo a capacidade de desenvolver, quase que em sua totalidade, um projeto. Porém, os *softwares* possuem uma área em que se destacam como projeto arquitetônico, estrutural, instalações, estimativa de custos, planejamento, questões de fabricação, entre outros (EASTMAN et al., 2014). A Figura 9 traz alguns exemplos de *softwares* disponíveis no mercado.

Figura 9 – Lista de Softwares BIM.

Foco	Software	Empresa	Website
Projeto Arquitetônico	Revit	Autodesk	autodesk.com.br
	ArchiCAD	Graphisoft	graphisoft.com/archicad
	VectorWorks	Nemetscheck	vectorworks.net/architect
	Bentley Architecture	Bentley	bentley.com
Análise e Compatibilização	Navisworks	Autodesk	autodesk.com.br
	Synchro	Synchro	verano.com.br/produtos/synchro
	Solibri	Nemetscheck	solibri.com.br
	Tekla BIMSight	Trimble	tekla.com/products/tekla-bimsight
Projeto Estrutural	EBERICK	AltoQi	altoqi.com.br/eberick
Projetos de Instalações	QiBuilder	AltoQi	altoqi.com.br/qibuilder
Colaboração BCF	BIMcollab	KUBUS	bimcollab.com

Fonte: autor.

Como nenhuma aplicação consegue realizar todas as tarefas associadas ao projeto e à produção de uma obra eficientemente (EASTMAN et al., 2014), é inevitável o uso de múltiplos *softwares*, principalmente em grandes projetos. Eventualmente, programas de diferentes empresas serão usados em um mesmo projeto, devido a funcionalidades e ferramentas disponíveis apenas em determinados *softwares*, facilitando a execução de tarefas específicas. Assim surge o grande problema de comunicação de dados entre programas de diferentes empresas.

Eastman et al. (2014), afirmam que a interoperabilidade consiste na possibilidade de múltiplas aplicações serem usadas para contribuir para um mesmo projeto, representando a necessidade de troca de dados entre os programas. Alves (2018) pontua as três maneiras mais comuns de se obter a interoperabilidade entre *softwares BIM*, sendo elas:

- *Softwares* desenvolvidos pelo mesmo fornecedor, em geral, conseguem ler diretamente os arquivos uns dos outros.
- *Softwares* que implementam uma *API (Application Programming Interface)* têm a capacidade de enviar informações para programas de diferentes fornecedores.
- *Softwares* que suportem padrões de troca de dados abertos. O padrão *CIS/2 (CIMSteel Integration Standards)* é um exemplo de padrão bem-sucedido, com

foco na indústria metálica. O *IFC (Industry Foundation Classes)* é um outro exemplo de padrão de troca de dados.

Para Jacoski (2003) apud Silva (2019), para que um software seja considerado interoperável, ele precisa ter as seguintes características:

- **Abertura:** deve existir a opção de criar aplicações integradas aos sistemas.
- **Troca:** o software deve ser capaz de trocar dados livremente entre os sistemas.
- **Uniformidade:** ter um padrão conhecido que possa ser usado.
- **Simplicidade:** ter o aprendizado facilitado.
- **Transparência:** capacidade de transformar toda a informação em apenas um formato.
- **Similaridade:** ter convenções similares entre os sistemas.

A falta de interoperabilidade entre os programas utilizados em um mesmo projeto acaba trazendo uma série de desvantagens. Segundo o *AIA (American Institute of Architects, 2009)* apud Ferreira (2011), verifica-se os seguintes problemas:

- Aumento de custos referentes a formação e requalificação em várias plataformas para compatibilização de projetos.
- Perda de acesso aos dados no futuro.
- Aumento de desperdícios de materiais, energia, dinheiro e tempo.
- Redução da produtividade devido a necessidade de reintrodução de dados e verificação de documentos.
- O desenvolvimento de ferramentas de análise e simulação, juntamente com interfaces para atender as rápidas mudanças da indústria, será comprometido.

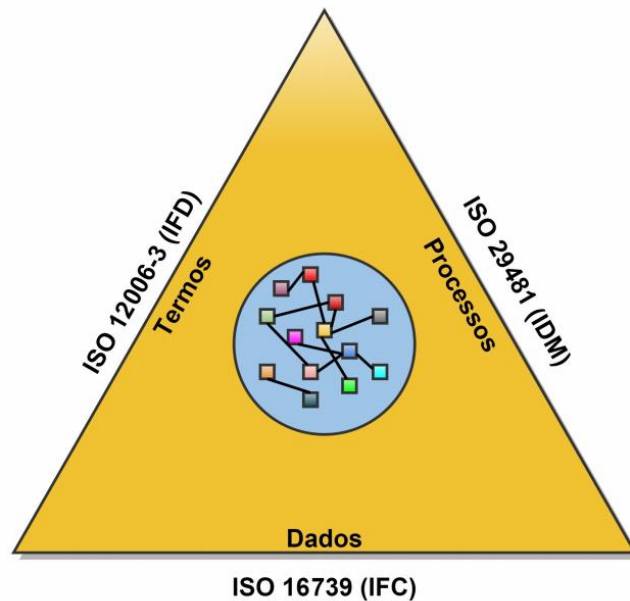
A necessidade de maneiras de realizar a interoperabilidade entre *softwares* BIM se tornou clara para todos os fornecedores. Assim, em 1995 a Autodesk organizou uma aliança privada com onze empresas que tinham um interesse em comum: investir no futuro da indústria da construção. Essa aliança tinha o objetivo de provar os benefícios da interoperabilidade e, após um ano de pesquisas, chegou à conclusão de que, além de viável, havia um grande potencial comercial (FREIRE, 2015).

A aliança, até então, privada percebeu que os padrões deveriam ser internacionais e abertos. Então, em 1996, foi estabelecida a *IAI* (*International Alliance for Interoperability*) que, posteriormente em 2008, mudou seu nome para *buildingSMART*. A nova organização se preocupava em implementar o *openBIM*, uma iniciativa que visa aplicar uma abordagem universal para colaboração de projetos (FREIRE, 2015). De acordo com a *buildingSMART* (©2021), os princípios que *openBIM* reconhece são os seguintes:

- **Interoperabilidade:** essencial para a transformação digital da indústria.
- **Aberto:** padrões neutros e abertos devem ser desenvolvidos para facilitar a interoperabilidade.
- **Confiável:** troca de dados dependerá de referências independentes de qualidade.
- **Colaboração:** o fluxo de trabalho é melhorado por formatos de dados abertos e ágeis.
- **Flexibilidade:** ter a escolha entre diversos tipos de tecnologias e aplicações gera valor para todos.
- **Sustentável:** padrões de dados interoperáveis de longo prazo geram garantias.

A *buildingSMART* também desenvolveu normas para auxiliar a indústria a se tornar mais eficiente. Essas normas são referentes a dados com formato livre (*Industry Foundation Classes – IFC*), processos (*Information Delivery Manual – IDM*) e terminologias (*International Framework of Dictionaries – IFD*) (FERNANDES, 2013). A Figura 10 demonstra os conceitos básicos do *openBIM* e as normas *ISO* relacionadas a eles.

Figura 10 – Pirâmide com os três conceitos básicos do openBIM.



Fonte: Freire (2015).

2.3.6. Industry Foundation Classes (IFC)

O *Industry Foundation Classes (IFC)* é um modelo de dados aberto desenvolvido pela *buildingSMART* com foco na indústria AEC (Arquitetura, Engenharia e Construção). Seu desenvolvimento tem o objetivo de criar um meio de intercâmbio de dados entre softwares, utilizando um grande conjunto de representações de dados consistentes de informações da construção (EASTMAN et al., 2014).

A *buildingSMART* (©2021) define o *IFC* como um padrão internacional aberto (ISO 16739-1:2018) feito para ser neutro e utilizável em diversas plataformas, dispositivos e interfaces diferentes. A organização também pontua que o esquema *IFC* é um modelo de dados padronizado que codifica os seguintes assuntos de uma forma lógica:

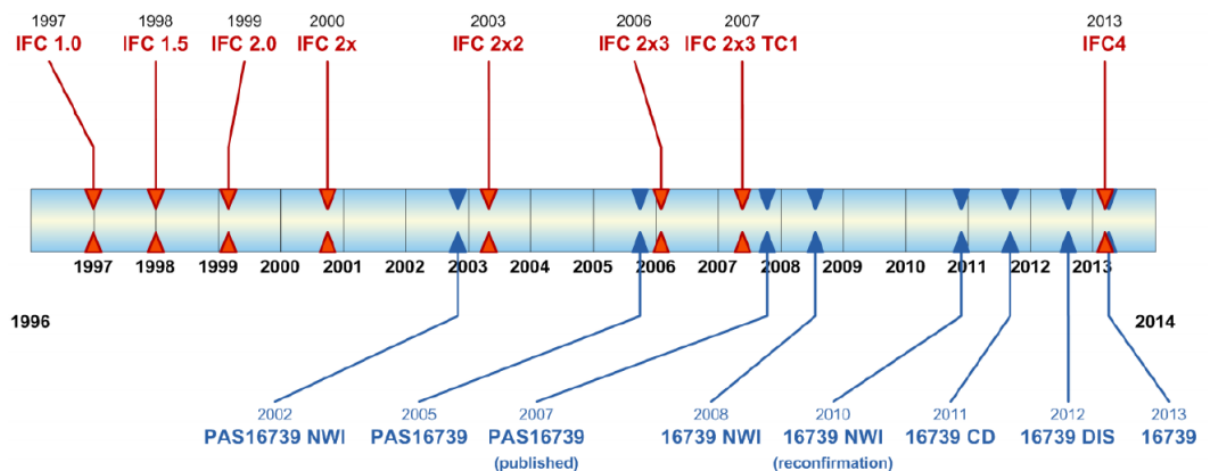
- **Identidade e semântica:** nomes, identificadores, tipos de objeto ou função.
- **Características ou atributos:** material, cor, propriedades térmicas etc.
- **Relações:** localização, conexões e hierarquia.
- **Objetos:** colunas, lajes etc.

- **Conceitos abstratos:** como performance ou custo.
- **Processos:** instalações e operações.
- **Pessoas:** designers, contratantes, fornecedores etc.

O *IFC* consegue definir componentes físicos de uma edificação, sistemas elétricos ou mecânicos, produtos manufaturados e até modelos abstratos de análise estrutural, modelos de análise energética, orçamentos, cronogramas e muito mais. O esquema tem a capacidade de descrever como uma instalação é usada, construída e operada (*buildingSMART*, ©2021).

O padrão de modelo de dados continua sendo desenvolvido, estando, atualmente, na versão *IFC4*. A cada atualização de versão, o padrão recebe adição de novas funcionalidades e melhorias, facilitando a integração entre diferentes *softwares*. O lançamento do *IFC2x* foi um marco para as versões, já que, desde então, o núcleo da especificação *IFC* permanece inalterado (PINHO, 2013). A Figura 11 ilustra os lançamentos do *IFC* ao longo do tempo.

Figura 11 – Cronologia do padrão IFC.



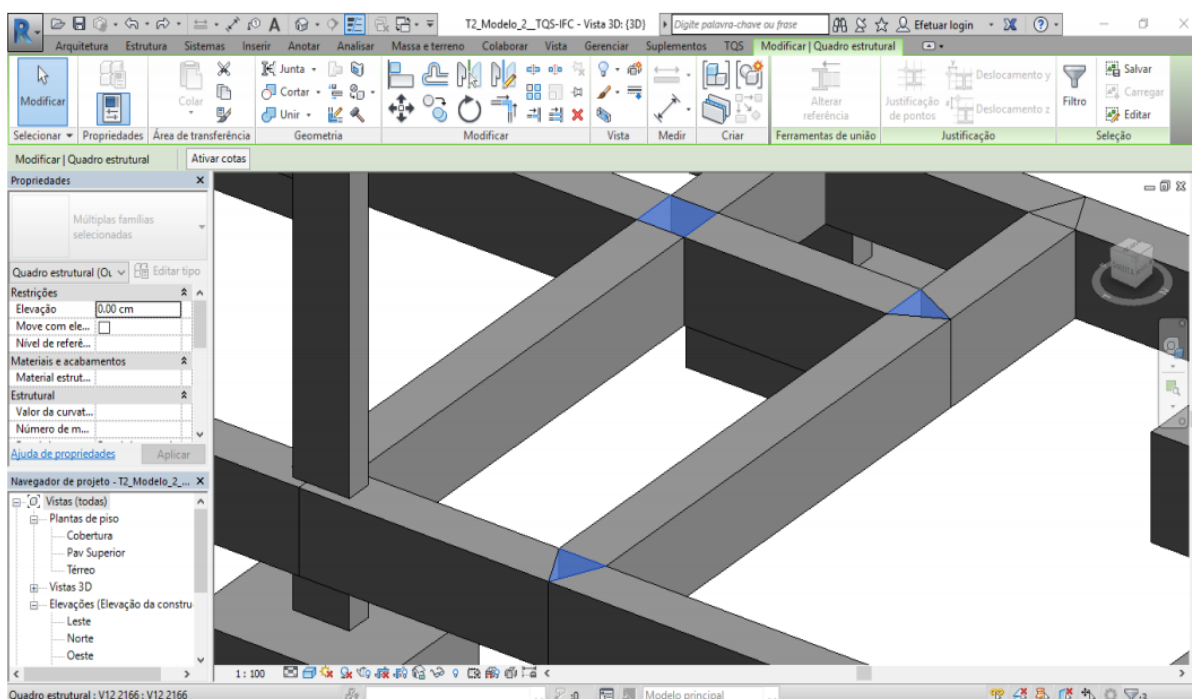
Fonte: Liebich (2013).

Apesar dos lançamentos e atualizações contínuas do padrão *IFC*, cabe as empresas proprietárias de *softwares BIM* implementarem o padrão nos seus programas.

Segundo Catelani (2016), ainda que a versão mais recente seja o *IFC4*, a versão predominantemente utilizada atualmente é a *IFC2x3*, lançada em 2006.

O desenvolvimento contínuo do padrão *IFC*, assim como sua adoção por parte dos *softwares BIM* é essencial para o crescimento e melhoria dos processos de interoperabilidade. A importação de modelos *BIM* por meio do *IFC* ainda apresenta diversos problemas. O estudo de Silva (2019) relatou problemas com a utilização do formato, alguns objetos perderam parâmetros e vinculações e outros elementos apresentaram falta de continuidade (Figura 12).

Figura 12 – Após importação do modelo IFC, elementos contínuos foram subdivididos em vários elementos e foram criados elementos de intersecção.



Fonte: Silva (2019).

O estudo de Barreto et al. (2020) também verificou a ocorrência de erros no processo de conversão do projeto original no modelo *IFC* para outros programas, os problemas observados variaram dependendo do *software* usado para visualizar o modelo. Foram observados desde erros pequenos relacionados a alteração de cores nos elementos,

até erros mais significativos como a exclusão de alguns pilares e vigas e alteração nas características externas (mudanças de materiais e acabamento) do modelo, como ilustrado na Figura 13 (BARRETO et al., 2020).

Figura 13 – Projeto sem acabamento. Modelo IFC visualizado pelo software B1Mer.



Fonte: Barreto et al. (2020).

Os problemas gerados com a importação do esquema *IFC* podem afetar o cronograma, planejamento e orçamento de uma obra, além de dificultar a identificação dos elementos contidos no projeto original. Assim, há uma grande contradição com a metodologia *BIM* e seu objetivo de facilitar o desenvolvimento de um projeto (BARRETO et al., 2020).

Para Alves (2018), em seu estudo sobre a interoperabilidade *BIM* em relação a projetos de estruturas, os formatos *IFC* ainda não permitem uma comunicação satisfatória entre *softwares BIM* para um projeto de estruturas, sendo a melhor solução uma comunicação de dados através de *APIs*.

2.3.7. Integração por meio de *Application Programming Interface (API)*

Uma *Application Programming Interface (API)* ou, em português, Interface de Programação de Aplicações, de uma forma geral, expõe um conjunto de dados e funções que facilitam a interação entre programas computacionais, permitindo a troca de informações (MASSÉ, 2011). Analogamente, uma *API* seria como um tradutor que intermedia uma interação entre duas pessoas que falam línguas diferentes (HERYANDI, 2018).

As *APIs* formam o núcleo do desenvolvimento moderno de *softwares*, sendo o trabalho de um engenheiro de *software* utilizar *APIs* para criar novos produtos, reutilizando códigos construídos por terceiros com o objetivo de economizar tempo e evitar reinventar a roda. Grandes empresas de tecnologia, como Google, Amazon e Yahoo!, já disponibilizam diversas *APIs* que oferecem serviços que podem ser usados por desenvolvedores para criar novas aplicações (JIN et al., 2018).

O conceito dessa tecnologia não é novo para a ciência da computação. Na década de 1960 desenvolvedores já haviam criado aplicações que permitiam o uso de funcionalidades de uma biblioteca sem a necessidade de saber seu código interno. Nas décadas de 1970 e 1980 surgiram *APIs* que funcionavam em computadores conectados em uma mesma rede. Somente durante os anos 1990, com o crescimento da *internet*, que as empresas começaram a padronizar o modo de construir e expor *APIs* (JIN et al., 2018).

No final dos anos 1990 e início dos anos 2000 vieram padrões mais abertos para acessar serviços remotos pela internet (*web APIs*), por exemplo: *Representative State Transfer (REST)* e *JavaScript Object Notation (JSON)*. Acessar serviços se tornou cada vez mais fácil e padronizado, sem dependências de linguagens de programação, popularizando ainda mais o uso de *APIs* (JIN et al., 2018).

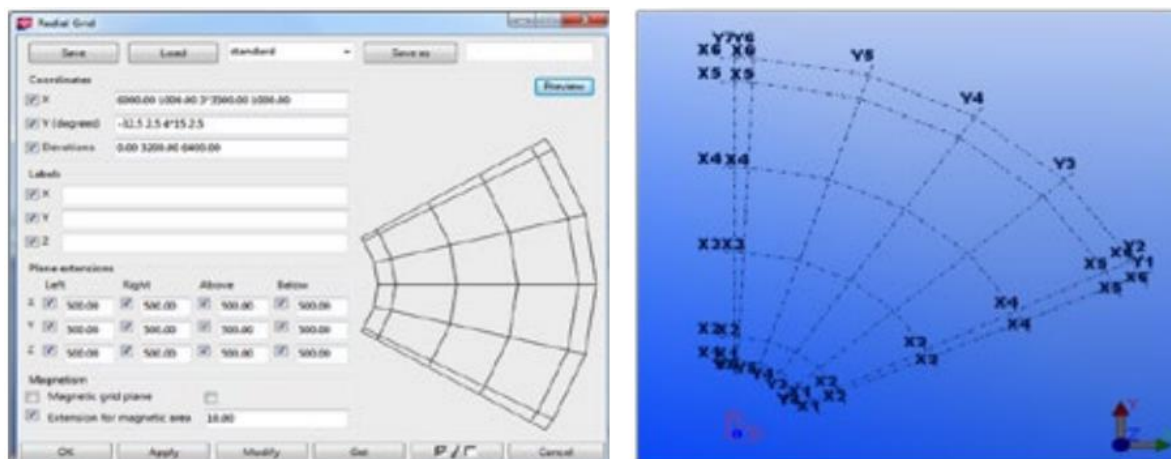
Ainda que *APIs* sejam feitas para trabalhar com outros programas, elas são criadas para serem entendidas e usadas por seres humanos, sendo a base para permitir a interoperabilidade entre grandes plataformas. A tecnologia também permite a integração de serviços de terceiros em uma aplicação, podendo servir de incentivo para a criação

de funções suplementares para um programa onde o desenvolvedor não teria a capacidade de investir na criação dessas funções (JIN et al., 2018).

Algumas soluções *BIM* possuem *APIs* que podem ser utilizadas como recurso de integração entre sistemas, fomentando a interoperabilidade em um projeto que exige o uso de diferentes *softwares BIM*. Geralmente o acesso dessas *APIs* é feito por meio de programação, podendo exigir ou não a intervenção de um profissional especializado (CATELANI, 2016).

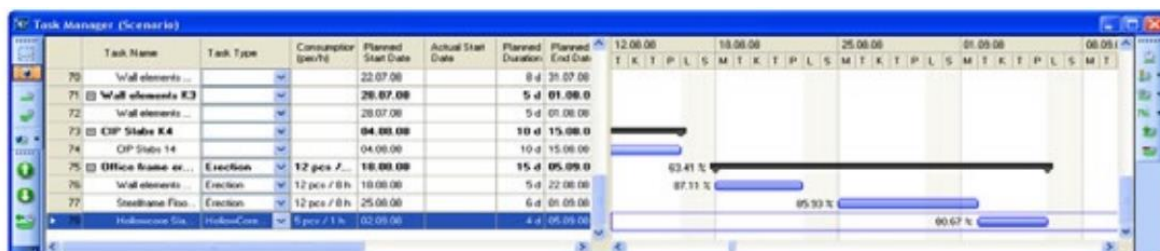
Recentemente os desenvolvedores de um *software BIM* vem fornecendo uma *API* específica para que sejam criados, por terceiros, *plug-ins* que estendem as funcionalidades do programa. Na maioria dos casos as linguagens de programação usadas são: C++, C# e VB. Integrações com o *Microsoft Excel* também são possíveis e muito utilizadas, especialmente para desenvolvimento de orçamento e estimativas de custos (CATELANI, 2016). As Figuras 14 e 15 ilustram exemplos de ferramentas criadas para *softwares BIM* por meio de uso de *APIs*.

Figura 14 – Grid Radial criado por meio de API no software Tekla Structures.



Fonte: Catelani (2016).

Figura 15 – Gerenciador de tarefas criado por meio de API no software Tekla Structures.



Fonte: Catelani (2016).

2.3.8. Benefícios da utilização da metodologia *BIM*

A indústria de AEC já percebeu ganhos significativos com a adoção do *BIM* quando comparado ao modelo tradicional de *CAD* 2D. A tecnologia é capaz de dar suporte e incrementar várias práticas no setor da construção civil, apesar da metodologia ainda estar em desenvolvimento, há diversas vantagens e mudanças esperadas com seu uso (EASTMAN et al., 2014).

Há benefícios na pré-construção para o proprietário, considerando que é possível desenvolver um modelo esquemático antes de gerar um modelo detalhado da construção, possibilitando vinculação a uma base de dados de custos podendo, então, verificar se a construção pode satisfazer os requisitos financeiros do proprietário (EASTMAN et al., 2014).

Em relação aos benefícios que podem ser observados na fase de projeto, Eastman et al. (2014) citam alguns pontos:

- **Visualização antecipada e mais precisa de um projeto:** o modelo 3D é gerado pelo *software* e pode ser usado para visualizar o projeto em qualquer etapa do processo.
- **Correções automáticas de baixo nível quando mudanças são feitas no projeto:** os objetos possuem relações estabelecidas por regras paramétricas, a alteração em uma regra gerará alterações em todos os objetos relacionados, automaticamente.

- **Geração de desenhos 2D precisos e consistentes em qualquer etapa do projeto:** há grande redução de tempo e número de erros associados com a geração dos desenhos, já que estes podem ser extraídos de forma precisa e consistente para qualquer parte do projeto.
- **Colaboração antecipada entre múltiplas disciplinas de projeto:** facilita o trabalho simultâneo de múltiplas disciplinas já que é possível trabalhar com mais de um modelo 3D de forma coordenada.
- **Verificação facilitada das intenções de projeto:** à medida que o modelo 3D é criado, áreas e materiais são quantificados, facilitando estimativas de custos de forma antecipada e precisa, auxiliando em projetos em que a intenção é definida quantitativamente.
- **Extração de estimativas de custo durante a etapa de projeto:** em qualquer etapa do projeto, um *software BIM* consegue fornecer uma lista detalha de quantitativos e espaços. Assim é possível manter todos os participantes conscientes das implicações dos custos associadas com dado projeto.
- **Incrementação da eficiência energética e a sustentabilidade:** a capacidade de vincular um modelo a ferramentas de análise energética proporciona diversas oportunidades para melhorar a qualidade da construção, permitindo, por exemplo, a verificação do uso de energia durante fases mais preliminares do projeto.

Já a respeito dos benefícios relacionados à construção e à fabricação, Eastman et al. (2014) pontuam o seguinte:

- **Sincronização de projeto e planejamento da construção:** é possível associar o modelo 3D a etapa de planejamento, criando um modelo *BIM* 4D. Assim, é possível efetuar uma simulação do processo de construção em qualquer ponto no tempo.
- **Descoberta de erros de projeto e omissões antes da construção (detecção de interferências):** todos os projetos podem ser colocados juntos e comparados, efetuando um processo de detecção de interferências e identificando conflitos que, antes, seriam encontrados apenas na obra.
- **Reação rápida a problemas de projeto ou do canteiro:** as consequências de uma modificação podem ser refletidas com precisão e rapidez no modelo, já que outros

objetos no projeto podem ser atualizados automaticamente. Modificações em um modelo *BIM* também minimizam erros que acontecem quando se utiliza sistemas baseados em papel.

- **Uso do modelo de projeto como base para componentes fabricados:** uma representação precisa dos componentes pode ser gerada a partir da importação do modelo para uma ferramenta *BIM* de fabricação.
- **Melhor implementação e técnicas de construção enxuta:** modelos *BIM* proporcionam a base para melhorias em planejamento e cronograma, devido ao fornecimento de um modelo de projeto preciso e dos recursos necessários para cada fase da construção. Assim reduz custos e permite melhor colaboração de trabalho.
- **Sincronização da aquisição de materiais com o projeto e a construção:** é possível adquirir materiais de fornecedores de produtos e subempreiteiros utilizando os levantamentos precisos de materiais e objetos que são gerados por um modelo *BIM* 3D.

Sobre benefícios na fase de pós-construção, Eastman et al. (2014) citam que é possível obter um melhor gerenciamento e operação das edificações, por meio das informações fornecidas pelo modelo de construção sobre todos os sistemas usados. A integração com sistemas de operação e gerenciamento de facilidades é outro benefício proporcionado, considerando as informações precisas sobre como os espaços e sistemas foram construídos, que fornecem um ponto de partida muito útil para o gerenciamento e a operação da construção.

Catelani (2016) também cita diversos pontos onde modelos que aplicam a metodologia *BIM* se destacam. Um deles é sobre a capacidade de executar construções complexas: as edificações estão tendo um grau de complexidade cada vez maior, tanto em suas formas quanto nas soluções tecnológicas utilizadas em seus subsistemas. Assim, ferramentas *BIM* se mostram adequadas para encarar esse cenário de condições desafiadoras.

A utilização de modelos *BIM* também pode influenciar e intensificar o uso da industrialização no setor da construção civil. Os *softwares* podem gerar componentes com

um grande grau de detalhes, garantindo um alto nível de confiabilidade e viabilizando o uso da pré-fabricação (CATELANI, 2016).

Apesar de modelos de interoperabilidade ainda estarem sendo desenvolvidos para melhorar a comunicação entre *softwares BIM*, o surgimento de novas ferramentas torna a interoperabilidade um fator cada vez mais necessário. Alves (2018) cita os principais benefícios de uma interoperabilidade ideal, sendo eles:

- Eliminar a necessidade de reintrodução de dados.
- Eliminar o uso de papel para comunicação entre projetos.
- Minimiza erros e falhas.
- Aumenta a rapidez da concepção do projeto.
- Permite a reutilização de informações.
- Diminui os custos de comunicação.
- Permite que os membros de uma equipe trabalhem em um mesmo projeto, aumentando a precisão do modelo já que todos trabalham sobre a mesma base.

As dimensões *BIM* ainda não conseguem ser aplicadas em sua totalidade de forma ideal. Porém, o estudo de Silva et al. (2019) mostrou diversos benefícios relacionados a otimização do processo construtivo com a utilização do *BIM 4D*. Um exemplo foi a redução em 7,6% no tempo destinado a execução de obra (CHEN et al., 2013, apud SILVA et al., 2019).

Já a pesquisa de Alrashed e Kantamaneni (2018) conseguiu demonstrar a eficácia do uso de modelos *BIM 5D* para a redução de custos de construção, já que foi possível evitar o desperdício de materiais e de tempo por meio de detecção de interferências, resultando em planejamento e cronograma precisos.

O setor de AEC tem aderido à metodologia *BIM* em várias partes do mundo. Essa tecnologia eleva a confiabilidade e a precisão dos projetos, além de romper paradigmas de produtividade. Qualquer empresa que queira se manter na indústria da construção deve se preparar para o futuro ao adotar modelos baseados em *BIM* (CATELANI, 2016).

2.3.9. Obstáculos relacionados à adoção do *BIM*

Existem diversas causas que bloqueiam a adoção da metodologia *BIM*, uma das principais está relacionada à inércia e à resistência às mudanças. A migração para sistemas *BIM* significa a troca de um modelo onde os envolvidos já estão acostumados para algo novo que altera como as atividades e os processos são executados. Para que essa mudança ocorra de fato são necessários cinco pontos: visão, capacitação, incentivos, recursos e um plano de ação (CATELANI, 2016).

A implementação de metodologia *BIM* também gera desafios relacionados a colaboração de equipes e mudanças na prática e no uso da informação. Esses modelos requerem adoção de métodos que permitam um compartilhamento adequado do modelo de informações, que servirá de base para todo processo de trabalho e colaboração, sendo uma das mudanças mais significativas que as empresas enfrentam (EASTMAN et al., 2014).

A dificuldade de entendimento e compressão também é outro fator que barra a implementação do *BIM*. Catelani (2016) afirma que é pouco intuitivo compreender o que é *BIM* e o significado de sua adoção para a indústria, também cita alguns pontos que atrasa a adoção do *BIM*:

- Não é fácil entender o que é *BIM* e qual seu significado.
- Não é fácil entender os potenciais benefícios que a metodologia pode proporcionar.
- Falta de entendimento que os custos altos, atualmente, são devidos a erros, retrabalhos, atrasos, demandas e processos.
- Proprietários e investidores não se deram conta que seriam os principais beneficiados da adoção do *BIM*.
- Bancos e agentes financiadores não perceberam que o modelo *BIM* garante maior precisão e confiabilidade, possibilitando prática de taxas mais baixas.

Em relação a especificidades e aspectos intrínsecos ao *BIM*, é possível afirmar que a adoção da metodologia requer esforço, aprendizado e investimento. Também, não é fácil comprovar o retorno sobre investimento com a utilização do *BIM*. Há ainda a

difficuldade de mensurar alguns dos principais benefícios oferecidos pela tecnologia, como o aumento da precisão nos projetos e no planejamento (CATELANI, 2016).

A pesquisa de Silva et al. (2019) relacionada a implementação de *BIM* 4D cita alguns pontos necessários para viabilizar a utilização efetiva do modelo 4D: utilização unânime de ferramentas *BIM*, padronizações de documentos e processos de modelagem, definição de fluxos de informação, entre outros. Carneiro e Maciel (2020) afirmam que a indústria de AEC apresenta uma grande falta de profissionais habilitados para utilizar as ferramentas *BIM*.

2.4. Programação e Automação

As grandes transformações no mundo, especialmente aquelas relacionadas ao comportamento da sociedade, estão diretamente relacionadas as mudanças tecnológicas, já que essas mudanças criam novas ferramentas e aprimoram procedimentos de trabalho, alterando profundamente os aspectos da vida e da cultura da humanidade (CHALINE, 2014, apud, SENA, 2019). As revoluções industriais estão fortemente ligadas a essas transformações, visto que sempre tiveram como foco avanços nos processos de produção, buscando sempre a mecanização e automação das operações (COELHO, 2016).

O mundo atualmente vive sua quarta revolução industrial, onde as novas tecnologias estão emergindo e se disseminando muito mais rápido do que em qualquer outra das revoluções. Esse processo está intimamente ligado com o desenvolvimento de tecnologias digitais e com a sofisticação e integração dos computadores. Os efeitos dessa revolução serão manifestados com todo seu potencial por meio da automação e criação de modelos e processos sem precedentes (SCHWAB, 2016).

2.4.1. Revoluções Industriais

A primeira revolução industrial teve início por volta de 1760 com a industrialização da Grã-Bretanha, devido ao foco no desenvolvimento industrial. Assim, desenvolveu-se

a indústria de base, representada pela metalurgia e siderurgia, tendo sua demanda ligada, principalmente, a construção das ferrovias nas primeiras décadas do século XIX (BRAICK e MOTA, 2007).

Um fator que marcou a primeira revolução industrial foi o uso do carvão como fonte de energia para indústria, para uso doméstico e como combustível para as locomotivas, permitindo o desenvolvimento do setor de transportes. Outro ponto marcante, foi a introdução de máquinas com o intuito de substituir o processo manual de produção (BRAICK e MOTA, 2007). Essa revolução deu um novo panorama aos meios de se locomover e produzir, visto que foi responsável pela dinamização do transporte de matérias-primas e pessoas (ARAUJO, 2018).

No final do século XIX e início do século XX se iniciou a segunda revolução industrial, tendo como seus principais pontos o advento da eletricidade e a possibilidade de produção em massa (SCHWAB, 2016). Também houve um aprimoramento significativo de técnicas já existentes, além de avanços nas áreas de química, elétrica e do aço (COELHO, 2016).

O setor de transportes teve grandes avanços com a expansão das estradas de ferro e com o crescimento da indústria naval. As invenções do telégrafo e telefone possibilitaram avanços no setor de comunicações. O motor à explosão também foi uma importante invenção que ocorreu nesse período (ROGGIA; FUENTES, 2016). Coelho (2016) cita que, durante esses anos, também surgiram os primeiros barcos de aço movidos por potentes motores a vapor, que revolucionaram o transporte de mercadorias.

Outro grande marco da indústria foi o Fordismo: um modelo de produção em massa idealizado por Henry Ford. Esse modelo barateou os artigos produzidos e ampliou o mercado consumidor, já que esse modo de produzir apresentava um baixo custo de fabricação e possibilitava um acúmulo de capital (ARAUJO, 2018). “A invenção e inovação andaram de mãos dadas nesta que foi a segunda revolução industrial.” (COELHO, 2016).

A terceira revolução industrial se iniciou na década de 1960 e se caracterizou pela proliferação e uso de semicondutores, automação e robotização em linhas de produção,

além do surgimento da internet, dos telefones móveis e da informação armazenada e processada digitalmente (COELHO, 2016). Essa revolução digital também trouxe o que viria a ser a base de toda a tecnologia da automação contemporânea: os computadores (ROGGIA; FUENTES, 2016).

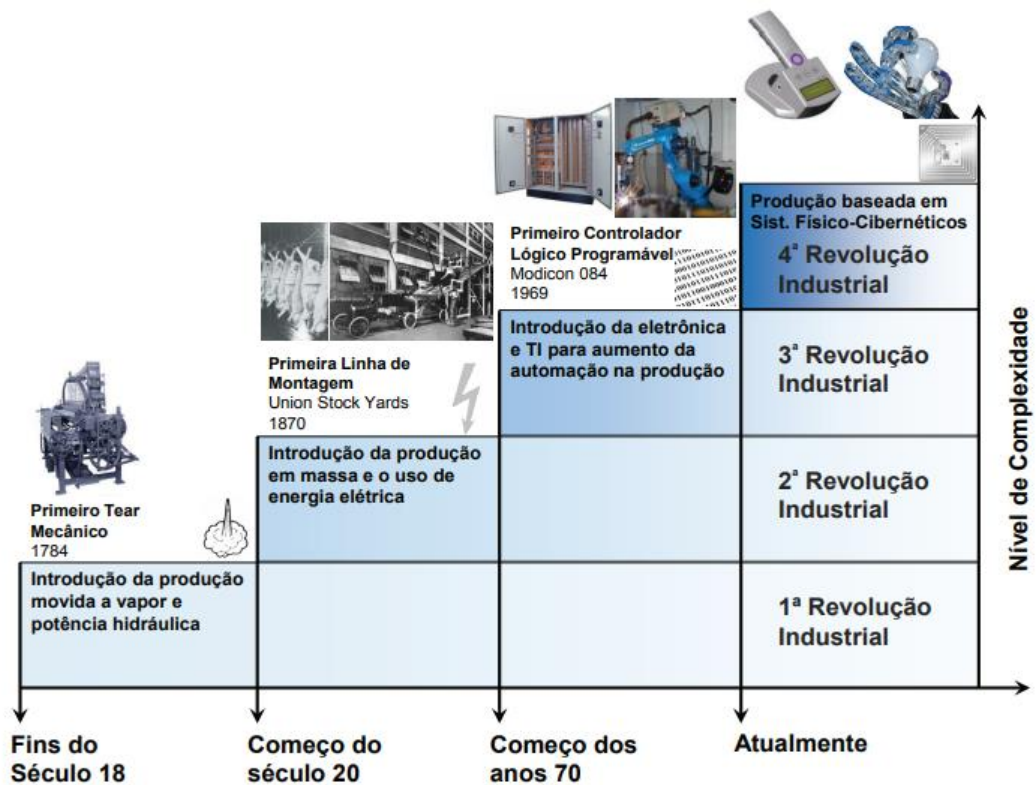
O surgimento do modelo chamado Toyotismo, em contraste com o Fordismo, aplicado no mercado automobilístico pelos japoneses também foi outro marco da terceira revolução industrial. Essa revolução também foi responsável por inovações nas áreas de biotecnologia, química moderna, telecomunicação e nanotecnologia (ARAUJO, 2018).

Para Schwab (2016), a quarta revolução industrial já começou e se baseia na revolução digital, se caracterizando por uma internet muito mais móvel e onipresente, por sensores cada vez menores e mais poderosos e por inteligências artificiais e modelos de *machine learning*, possibilitando a criação de um mundo onde sistemas virtuais e físicos consigam cooperar globalmente de um modo flexível, permitindo o surgimento de novos modelos operacionais.

Em contraste com a terceira revolução industrial, os computadores estão se tornando cada vez mais sofisticados e integrados, sendo catalisadores da transformação da sociedade e da economia global (SCHWAB, 2016). Araujo (2018), afirma que as mudanças econômicas, sociais e culturais devido a atual revolução tecnologia tem proporções tão grandes que sua previsibilidade é extremamente difícil.

Schwab (2016) pontua que a quarta revolução industrial é fundamentalmente diferente das outras três, visto que há ondas simultâneas de avanços e descobertas em diversas áreas diferentes, possibilitando a fusão e interação dessas novas tecnologias, seja fisicamente, digitalmente e, até mesmo, biologicamente. A Figura 16 traz um resumo do que foi alcançado em cada revolução industrial.

Figura 16 – Progressão das revoluções industriais.



Fonte: Rodrigues, Jesus e Schützer (2016).

2.4.2. Automação

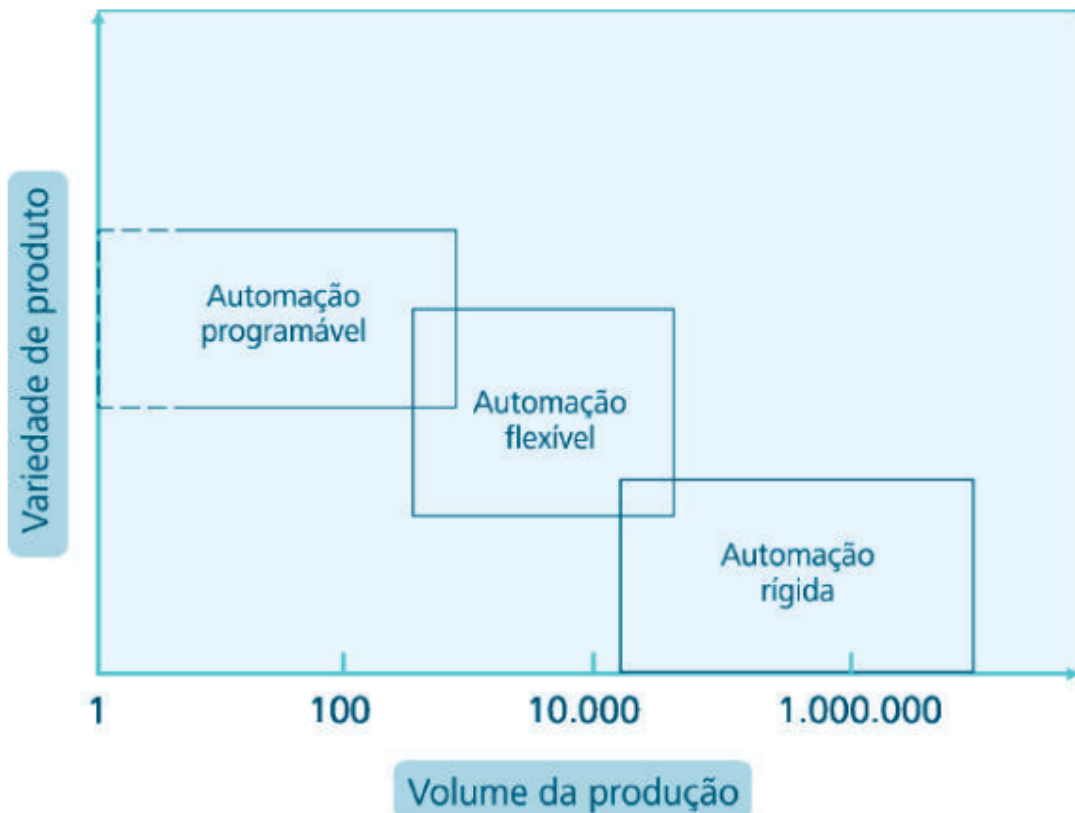
Araujo (2018) traz a definição de automação como sendo a utilização de técnicas computadorizadas ou mecânicas por meio de um sistema, com o intuito de dinamizar e otimizar os processos produtivos dos mais variados setores da economia. Já para Roggia e Fuentes (2016), automação é um sistema de controle, sem a necessidade da intervenção do homem, que, possui mecanismos que verificam sua própria operação, efetuando medições e realizando correções.

A automação não deve ser confundida com mecanização, sendo o último relacionado a utilização de máquinas para executar determinada tarefa em detrimento do esforço físico, em contraste com a automação, que possibilita a execução de tarefas por meio de máquinas que são controladas automaticamente (ROGGIA; FUENTES, 2016).

De acordo com Roggia e Fuentes (2016), os sistemas de automação industrial também podem ser classificados de acordo com seu grau de flexibilidade, da seguinte forma:

- **Automação rígida:** focada na fabricação de um produto específico. Possui altas taxas de produção (Figura 17), porém apresenta inflexibilidade de equipamentos na acomodação da variedade de produção.
- **Automação programável:** é controlado por um programa que é interpretado pelo sistema. Apresenta suporte a diferentes configurações de produtos e diferentes programas podem ser utilizados para fabricação de novos produtos. Como ponto negativo possui um volume de produção baixo.
- **Automação flexível:** o equipamento deve ser programado para produzir uma variedade de produtos, tentando conciliar características da automação rígida e outras da automação programável. Os produtos podem ter características ou configurações diferentes, porém são mais limitados do que os fabricados pela automação programável.

Figura 17 – Volume de produção nos tipos de sistemas de automação industrial.



Fonte: Roggia e Fuentes (2016).

Além da sua utilização industrial, com a fabricação de produtos em massa, a automação também consegue trazer benefícios para diversos outros setores da economia. Para o mercado de Arquitetura e Engenharia, a automação tende a trazer taxas maiores de produtividade e facilita a mitigação de problemas constantes do mercado (ARAUJO, 2018).

Araujo (2018) cita em seu trabalho, diversos tipos de inovações que a automação pode trazer no mercado da construção civil. Entre eles, é possível destacar:

- **Drones:** são veículos aéreos não tripulados (Figura 18) e comandados com a ajuda de um controle remoto, podendo ser manuseados até mesmo por um telefone celular. Conseguem acessar pontos remotos com segurança e coletar imagens com o objetivo de analisar o local que deve ser estudado.

Figura 18 – Drone.



Fonte: Araujo (2018).

- **Impressão 3D:** tem o objetivo de agilizar o processo de execução e montagem de produtos. Consegue otimizar a velocidade de construção em empreendimentos, podendo ser usada para produzir até mesmo grandes elementos, como paredes e instalações.
- **Máquinas de reboco de paredes automática:** são máquinas que substituem o processo de execução de reboco manual em paredes. O equipamento (Figura 19) elimina a necessidade de efetuar chapisco e produz um revestimento compacto e uniforme, eliminando os problemas quanto a variabilidade da qualidade da execução.

Figura 19 – Máquina de reboco de paredes automática.



Fonte: Araujo (2018).

- **Assentadores de argamassa:** é uma nova técnica automatizada que visa agilizar o processo manual, apresentando um rendimento de execução muito superior e uma qualidade e nivelamento incomparáveis. Também possui grande facilidade de implantação e desenvolvimento do trabalho.
- **Automação predial:** um sistema predial possibilita a integração de equipamentos e áreas, coletas e análise de dados e segurança de informações, colaborando com a gestão de um edifício. Construções com sistemas de automação possuem a capacidade de apresentar altos níveis de eficiência e sustentabilidade.

Todos esses sistemas de automação, seja industrial ou até mesmo as inovações mencionadas para construção civil, envolvem a utilização de um *software* ou de um aparelho eletrônico digital chamado Controlador Lógico Programável (CLP) para administrar os processos envolvidos na automação. Portanto, os computadores e o conhecimento em como programá-los constituem a base de toda a tecnologia da automação

contemporânea, tendo exemplos de sua utilização nos mais diversos setores da economia (ROGGIA; FUENTES, 2016).

2.4.3. Programação de computadores

Uma busca constante na história da humanidade foi por meios de criar de ferramentas que auxiliassem no trabalho, diminuindo esforço e economizando tempo. Com esse contexto, foram criados os computadores: máquinas versáteis, úteis e seguras, com a capacidade de armazenar dados, imprimir relatórios, gerar gráficos, realizar cálculos e muitas outras coisas (ASCENCIO; CAMPOS, 2012).

Os computadores são capazes de tomar decisões lógicas milhões e até bilhões de vezes mais rápido que uma pessoa. Eles consistem em dispositivos físicos chamados de *hardware* (teclado, telas, *mouse*, memória etc.) e em programas que executam nos computadores, os chamados *softwares*. Esses programas são um conjunto de instruções que possibilitam o computador processar dados e essas instruções, por sua vez, são especificadas por pessoas chamadas de programadores (DEITEL; DEITEL, 2006).

No processo de criação de um software, um programador deve escrever o conjunto de instruções a serem realizadas pelo computador em uma linguagem que seja compreensível tanto para o computador quanto para o programador. Existem centenas das chamadas linguagens de programação, algumas delas são diretamente compreensíveis pelo computador e outras necessitam de um passo intermediário de tradução. É possível dividir as linguagens em três tipos gerais: de máquina, *assembly* e de alto nível. (DEITEL; DEITEL, 2006).

A linguagem de máquina é uma linguagem na qual qualquer computador pode entender diretamente, sendo uma espécie de “linguagem natural” dos computadores, definida pelo design de *hardware*. Elas são dependentes de máquina, ou seja, só podem ser utilizadas em um tipo de computador. O grande problema desse tipo de linguagem é a sua complexidade para serem escritas pelos programados (DEITEL; DEITEL, 2006).

Considerando a dificuldade e tendência a erros da escrita da linguagem de máquina, os programadores começaram a utilizar abreviações em inglês para representar operações elementares, assim nasciam as linguagens *assembly*. Programas tradutores, chamados *assemblers*, também foram criados, com o objetivo de converter os programas em linguagem *assembly* para linguagem de máquina (DEITEL; DEITEL, 2006).

Ainda que as linguagens *assembly* trouxeram grandes facilidade para o mundo da programação, muitas instruções eram necessárias para realizar simples tarefas. Com o intuito de acelerar o processo de programação e criar instruções únicas que realizaram tarefas substanciais, foram criadas as linguagens de alto nível. Linguagens de alto nível possuem uma sintaxe próxima à da língua inglesa e necessitam de programas tradutores, chamados de compiladores, para converter os códigos em linguagem de máquina. Alguns exemplos dessas linguagens são: C, C++, Visual Basic, C# e Java (DEITEL; DEITEL, 2006).

A Figura 20 exemplifica as diferenças entre os códigos escritos nos três tipos de linguagem. É perceptível que as linguagens de mais baixo nível (de máquina e *assembly*) possuem uma complexidade maior, podendo ser mais propensas a erros de escrita por parte dos programadores e já uma linguagem de alto nível consegue ser visivelmente mais simples e compreensível, portanto, mais intuitiva e acessível (SENA, 2019).

Figura 20 – Diferenças entre os tipos de linguagens de programação.

Programa para somar remuneração bruta (remunBruta = remuneração básica + horas extras)	
Linguagem de máquina	+1300042774 +1400593419 +1200274024
Linguagem assembly	LOAD REMUNBASICA ADD HORASEXTRAS STORE REMUNBRUTA
Linguagem de alto nível	remunBruta = remunBasica + horasExtra

Fonte: Sena (2019).

Para Ascencio e Campos (2012) a primeira etapa no desenvolvimento de um programa é a análise do problema a ser solucionado, para definir os dados de entrada, o processamento e os dados de saída. Então deve ser elaborado um algoritmo que deverá ser transformado em códigos de uma linguagem de programação (ASCENCIO; CAMPOS, 2012).

Assim, um algoritmo servirá de base para o desenvolvimento de softwares. É possível definir algoritmo como uma sequência lógica e finita de instruções que serão seguidas a fim de chegar na resolução de um problema. Além de seu uso ligado à área de ciências exatas, no dia a dia os algoritmos são usados de forma intuitiva, quando, por exemplo, uma pessoa segue uma receita culinária, que nada mais é do que uma sequência de instruções que devem ser realizadas para executar uma tarefa (PUGA; RISSETTI, 2009).

Após a codificação do algoritmo, a forma como o código é estruturado pode ser dividida em alguns tipos. Segundo Puga e Risettti (2009), as formas de estruturação são: programação linear, programação estruturada e programação orientada a objetos, podendo ser definidas da seguinte forma:

- **Programação Linear:** nesse tipo de estruturação, os programas obedecem a uma sequência de passos executados consecutivamente, tendo início e fim definidos. Foi um princípio utilizado nas primeiras línguas de programação, porém programas lineares são extensos e difíceis de ser compreendidos e desenvolvidos. Assembly e Basic são exemplos de linguagens de programação que se apoiam nesse tipo de estruturação.
- **Programação Estruturada:** o conceito desse tipo de estruturação se baseia na filosofia de dividir um problema em partes e atacá-las separadamente. A divisão de tarefas na programação estruturada recebe o nome de modularização, onde o programa é dividido em partes ou módulos que executam tarefas específicas. Procedimentos ou funções, com a capacidade de receber valores, são ferramentas que visam a execução de determinada tarefa.
- **Programação Orientada a Objetos:** esse tipo de programação revolucionou os conceitos de projeto e desenvolvimento de sistemas já existentes. A programação

orientada a objetos busca fazer um paralelo do mundo real com o mundo da programação, assim um programa será formado por objetos que interagem entre si e possuem atributos, apresentando comportamento e característica próprios, podendo realizar operações. Portanto os dados estão relacionados aos atributos dos objetos, assim como os procedimentos estão relacionados com as operações.

Sena (2019) também cita a programação visual como uma quarta forma de estruturação. Esse tipo de programação traz uma interface mais intuitiva para a criação de códigos a partir de nós de programação, se conectando a outros utilizando entradas e saídas de dados. Exemplos de programas que utilizam a programação visual são o Dynamo e o Grasshoper (SENA, 2019).

2.4.4. Linguagem Python

A criação da linguagem Python começou no final do ano 1989, por meio do pesquisador Guido van Rossum da *National Research Institute for Mathematics and Computer Science in Amsterdam (CWI)*. Guido precisava de uma linguagem de script de alto nível para realizar tarefas administrativas para seu grupo de pesquisa, porém as linguagens de programação disponíveis não satisfaziam o que precisava. Assim, pela combinação de diversos elementos de linguagens já existentes, juntamente com a capacidade de ser extensível por meio de classes e interfaces de programação, foi criada a linguagem Python (DEITEL et al., 2002).

Python é uma linguagem de programação de propósito geral, orientada a objetos e tem seu código aberto. Também possui uma sintaxe simples e legível, tendo seu foco em diminuir o tempo de desenvolvimento de código. É utilizada por grandes empresas como Google e Youtube e até mesmo pelo NASA, podendo criar aplicações nas mais diversas áreas, desde desenvolvimento *web* até ciência de dados (LUTZ, 2011).

Lutz (2009) cita que, após mais de 12 anos ensinando Python, foi possível perceber as seguintes características, como sendo os principais fatores observados pelos usuários da linguagem:

- **Qualidade de *software*:** Python traz suporte a mecanismos avançados de reuso de *software*, como a orientação de objetos. O código é focado em ser legível e coerente e, portanto, reutilizável e sustentável. Assim, a linguagem se destaca de outras ferramentas no mundo da programação.
- **Produtividade do desenvolvedor:** o código em Python roda imediatamente, sem a necessidade de compilação e, usualmente, é um terço a um quinto menor do que códigos escritos nas linguagens C++ ou Java (Figura 21). Desta maneira, há grande destaque quanto a produtividade e velocidade na programação.
- **Portabilidade de programa:** de uma forma geral, fazer com que o código em Python rode em sistemas operacionais diferentes, necessita apenas que haja uma cópia do código entre as diferentes máquinas
- **Bibliotecas de suporte:** Python possui uma biblioteca padrão que traz uma grande coleção de funcionalidades. A biblioteca padrão consegue auxiliar na criação de aplicações de todo nível, desde comparações de texto até aplicações de rede. Também deve ser mencionada a grande quantidade de bibliotecas de terceiros, estendendo ainda mais o suporte ao desenvolvimento de *software*.
- **Integração de componentes:** *scripts* escritos em Python possuem diversos mecanismos de integração, permitindo a comunicação com outras partes de uma mesma aplicação e até de interfaces diferentes, sendo possível interagir com dispositivos e redes. Atualmente, um programa em Python tem a capacidade de se integrar até com diferentes linguagens, podendo utilizar bibliotecas das linguagens C e C++, por exemplo.

Figura 21 – Comparação do código escrito em Python com C++ e Java.

Linguagem Utilizada	Programa: declara duas variáveis numéricas 'a' e 'b' e então imprime o resultado da soma entre os dois valores.
C++	<pre>#include <iostream> using namespace std; int main () { int a = 1; int b = 2; cout << "Soma de " << a << " + " << b << " = " << a + b << endl; return(0); }</pre>
Java	<pre>public class Main { public static void main (String[] args) { int a = 1; int b = 2; System.out.format("Soma de %d + %d = %d", a, b, a+b); } }</pre>
Python	<pre>a = 1 b = 2 print(f'Soma de {a} + {b} = {a+b}')</pre>

Fonte: autor.

Com a linguagem Python é possível criar aplicações para as mais diversas áreas computacionais. Uma das áreas em que Python tem tido um grande destaque é a de ciência de dados. A profissão de cientista de dados envolve a junção de habilidades em três áreas diferentes: conhecimentos em estatística, para modelar e resumir um conjunto de dados, as habilidades em ciência da computação, para criar e utilizar algoritmos que possam guardar, processar e visualizar esses dados de forma eficiente e, por fim, o domínio do assunto a ser analisado, necessário para formular as perguntas corretas e, portanto, trabalhar corretamente com os dados a serem estudados (VANDERPLAS, 2016).

Outra área que cresce cada vez mais com a utilização de Python é a de *machine learning* (ML). Os estudos de ML são sobre extrair informação de dados, tendo seu campo de pesquisa como uma combinação entre estatística, inteligência artificial, ciência

da computação e análise preditiva. Muitos dispositivos possuem algoritmos de *ML* em seu núcleo, sendo alguns exemplos de aplicação: recomendações automáticas de produtos para comprar, reconhecimento facial, detecção de atividade fraudulenta em cartões de crédito, determinar se um tumor é benigno baseando-se em sua imagem médica e diversos outros exemplos (MÜLLER; GUIDO, 2016).

A linguagem Python não foi criada com o propósito de servir para análise de dados ou computação científica. Isso só foi possível graças ao desenvolvimento e disseminação de bibliotecas de terceiros, tornando Python uma ferramenta de primeira classe para análise e visualização de grandes conjuntos de dados (VANDERPLAS, 2016). Python possui bibliotecas que permitem o carregamento de dados, visualização, estatística, processamento de linguagem natural, processamento de imagem e muito mais. Outra grande vantagem é a possibilidade de interagir diretamente com o código, já que análise de dados e *ML* são processos fundamentalmente iterativos (MÜLLER; GUIDO, 2016).

2.5. Integração Revit – Python

Para Labib (2017), a programação de computadores tem a capacidade de auxiliar arquitetos e engenheiros em superar os desafios relacionados a utilização de novas ferramentas digitais. O conhecimento em programação é crucial para tornar o processo de modelagem mais eficiente, podendo facilitar a organização, análise e visualização de simulações complexas de dados para a eficiência da edificação. Assim, programas customizados são essenciais para a criação de obras de alto desempenho (LABIB, 2017).

Considerando a estrutura de modelos *BIM* como sendo um grande conjunto de dados, devido aos objetos serem paramétricos, os *softwares* que aplicam a metodologia se tornam perfeitamente adequados aos processos inerentes à programação. Por meio da utilização da automação, conhecimentos em programação dão aos usuários de *softwares BIM* a capacidade de tornar seus processos mais rápidos e produtivos (SENA, 2019).

2.5.1. Autodesk Revit e sua API

O Revit é um *software* que aplica a metodologia *Building Information Modeling*, foi criado e é distribuído pela empresa Autodesk. De acordo com a Autodesk (©2021b), o Revit pode ser utilizado durante todo o ciclo de vida de uma edificação, desde seu design inicial, visualização e análise, até a fabricação e construção. É capaz de criar modelos 3D com precisão, além de permitir a colaboração interdisciplinar das diversas áreas presentes na elaboração de uma construção (AUTODESK, ©2021b).

Além de todas suas funcionalidades nativas, o Revit também permite a integração de novas ferramentas por meio de *plugins*. Os *plugins* são uma forma de expandir o uso de um *software*. Os fabricantes permitem que desenvolvedores criem funcionalidades que não estavam presentes no programa original (SENA, 2019). A Autodesk disponibiliza uma loja de aplicativos, onde é possível encontrar aplicações criadas por terceiros que melhoram a produtividade de um *software* Autodesk (AUTODESK, ©2021a).

Outro método de agregar funcionalidades ao Revit é por meio de sua *API*. De acordo com a Autodesk (©2021c), a empresa disponibiliza uma *API* rica e poderosa, capaz de automatizar tarefas e estender a funcionalidade do Revit em todas suas áreas. A *API* permite programação com qualquer linguagem da plataforma de desenvolvimento .NET, como VB.NET, C# e C++/CLI. Além disso, é disponibilizado documentação e exemplos de códigos para auxiliar com o uso da *API* (AUTODESK, ©2021c).

A *API* do Revit permite a criação de programas que podem ser integrados ao Revit. Entre os diversos usos da *API*, é possível citar as seguintes (AUTODESK, ©2021c):

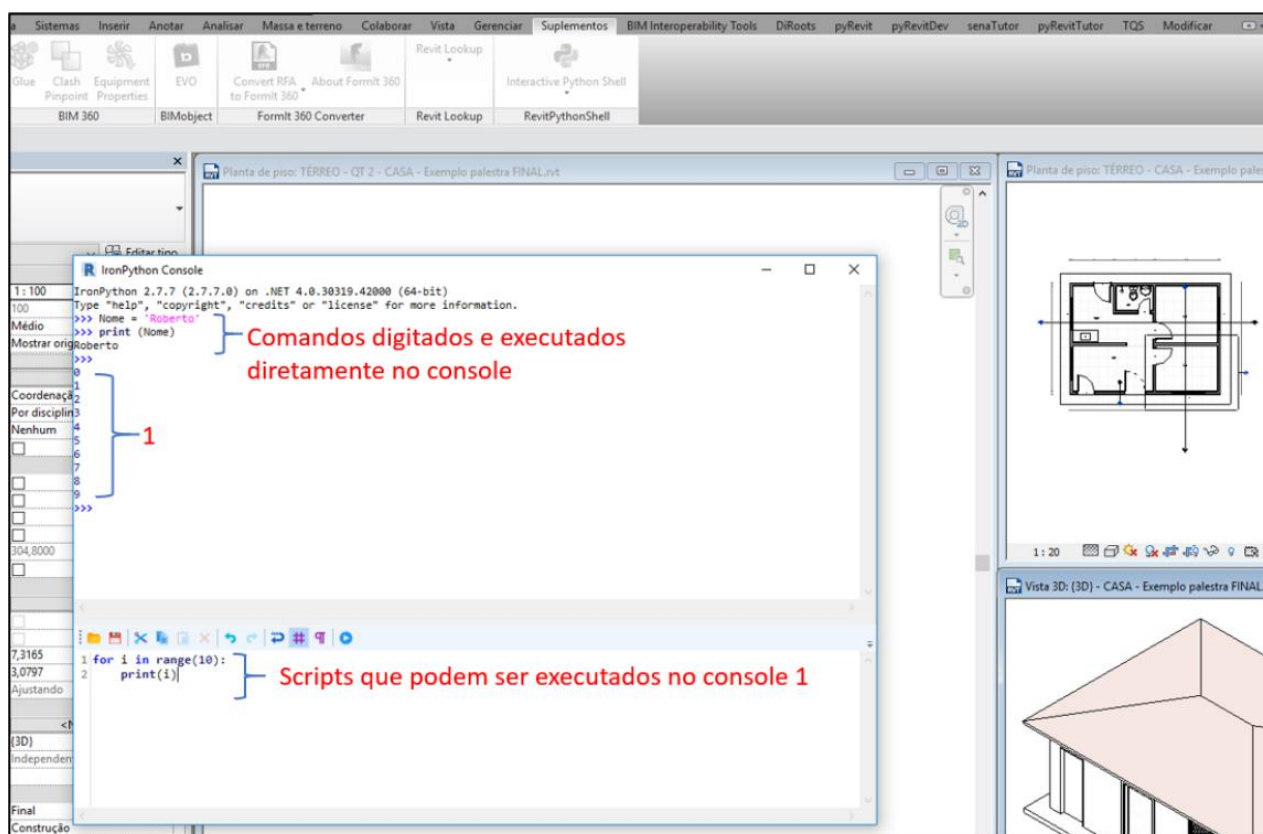
- Criar macros e *add-ins* que automatizam tarefas repetitivas.
- Verificar automaticamente se o projeto segue os padrões pré-definidos.
- Extrair dados do projeto para análise e gerar relatórios.
- Importar dados externos para criar novos parâmetros ou elementos.
- Integrar outras aplicações ao Revit.

2.5.2. Revit Python Shell (RPS)

Para possibilitar a implementação de *plugins* na linguagem Python, o Autodesk Revit adota um pacote de programação Python escrito em C#, desenvolvido para a utilização na plataforma .NET, chamado *IronPython* (SENA, 2019). O *plugin Revit Python Shell (RPS)* adiciona um interpretador de *IronPython* para o Autodesk Revit, permitindo a criação de *plugins* para Revit utilizando Python. Além disso, também traz um *shell* interativo que possibilita a visualização dos resultados do código conforme ele é escrito, sendo uma ótima forma de explorar a *API* do Revit e criar novos *plugins* (THOMAS, [s.d.]).

De acordo com Sena (2019), o *Revit Python Shell*, devido a sua forma de acesso aos dados, traz maior facilidade para a utilização do *API* do Revit quando comparado a métodos que utilizam consultas codificadas. O *RPS* possui uma interface que disponibiliza dois campos, o primeiro onde comandos serão executados imediatamente, já no segundo um trecho de código pode ser escrito e, posteriormente, executado (SENA, 2019). A Figura 22 ilustra a interface do *RPS*.

Figura 22 – Interface do Revit Python Shell.

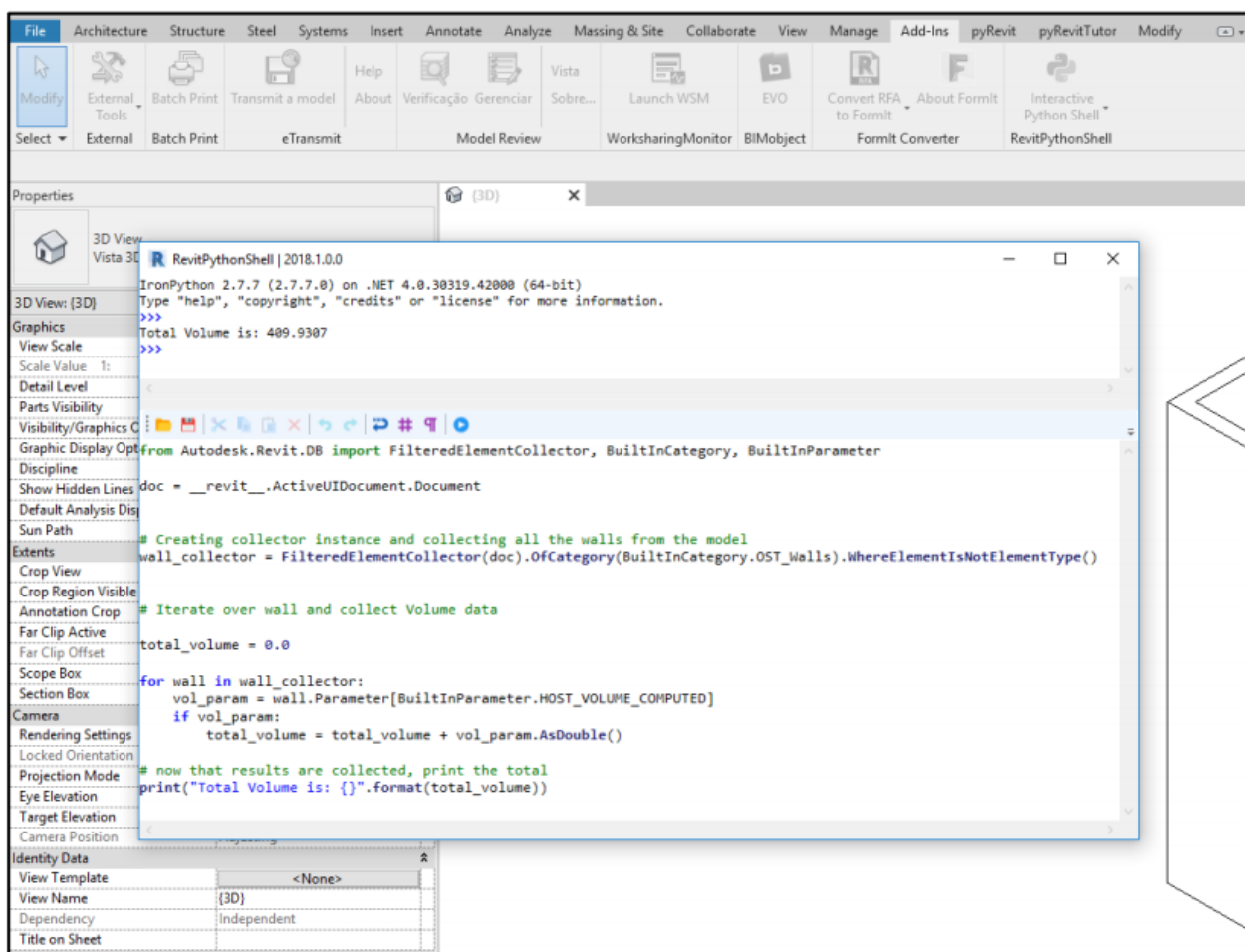


Fonte: Sena (2019).

Por permitir o acesso e a exploração dos dados presentes em um modelo *BIM* via *API* do Revit, o *RPS* possibilita personalização do *software* por meio da automatização de processos repetitivos, checagem de padrões pré-definidos e até mesmo a extração e estudo dos dados fora do escopo do Revit, assim evitando cada vez mais erros humanos e otimizando o tempo. O repositório do GitHub, referente ao *RPS*, também contém diversos exemplos de códigos e referências que auxiliam na utilização do *plugin* (THOMAS, [s.d]).

Sena (2019) demonstra um exemplo de código que ilustra a aplicação do *Revit Python Shell*. Nesse exemplo, o código, ao ser executado, utiliza a *API* disponibilizada pelo Revit para calcular o volume total das paredes inseridas no modelo. Na Figura 23 é possível visualizar o código escrito em Python no *RPS* e seu resultado.

Figura 23 – Exemplo de *script* que determina o volume das paredes do modelo.



Fonte: Sena (2019).

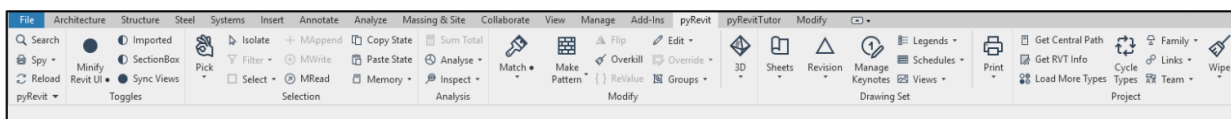
2.5.3. *pyRevit*

De acordo com a descrição presente no repositório do GitHub (EHSAN IRAN-NE-JAD, [s.d.]), o *pyRevit* é uma ferramenta que permite a automação de processos e criação de *addons* dentro do ambiente Revit, juntamente com a utilização de sua *API*. O *plugin* também traz um extenso conjunto de ferramentas que demonstram suas capacidades como ambiente de desenvolvimento.

Após a instalação do *pyRevit*, um conjunto de *scripts* são disponibilizados gratuitamente e uma nova aba (Figura 24) é adicionada ao Revit, onde estarão agrupados os ícones referentes aos *scripts* que poderão ser executados diretamente no Revit (SENA, 2019). Também é possível acessar o código dos *scripts* que acompanham a instalação

do *pyRevit*, possibilitando o aprendizado na utilização do *IronPython*, juntamente com a *API* do Revit. Uma maneira fácil de acessar o *script* de um comando é clicar em seu ícone segurando a tecla *alt* (SENA, 2019).

Figura 24 – Aba *pyRevit* com novos comandos, disponibilizada após instalação do plugin.



Fonte: Sena (2019).

O *pyRevit* não necessita que seus scripts sejam rodados em um console de programação, em contraste com o *RPS*, sendo este último ideal para testar códigos, devido a sua flexibilidade e respostas em tempo real. Já o *pyRevit*, consegue tornar o processo mais ágil ao permitir a criação de um ícone que rodará um *script*, já testado, apenas com um clique, automatizando até mesmo a inserção dos códigos (SENA, 2019).

Sena (2019) utilizou parâmetros estipulados pela lei municipal de Ribeirão Preto para elaborar um algoritmo que checa se as áreas dos cômodos estão de acordo com o que foi estabelecido. Segundo Sena (2019), o *RPS* foi utilizado primeiramente para realização de testes e estudos e, então, com o *pyRevit* foi criada uma nova aba no *software* Revit, que contém um comando que executa o código. A Figura 25 demonstra o código elaborado por Sena (2019) e a Figura 26 mostra o resultado da execução do código ao clicar no botão “Lista”.

Figura 25 – Rotina elaborada em Python por Sena (2019).

```

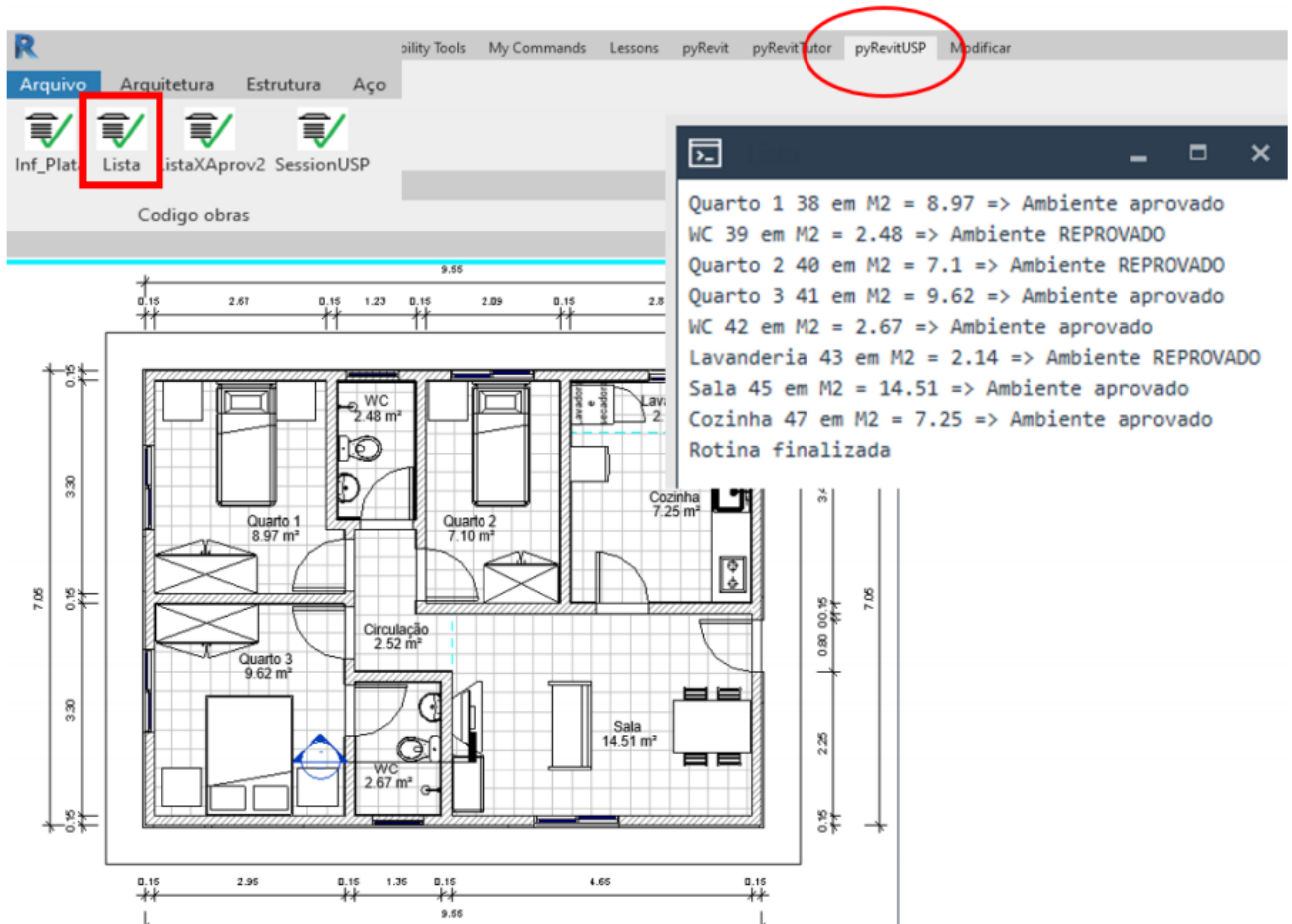
1  """Mestrado USP - Rotina Python - September DE 2019
2  PAULO CÉZAR FEIXOTO DE SENA"""
3
4  # CLASSE DE OBJETOS QUE SÃO IMPORTADAS VIA API DO REVIT
5  from Autodesk.Revit.DB import Transaction, Element
6  from Autodesk.Revit.DB import FilteredElementCollector
7  from Autodesk.Revit.UI import TaskDialog
8  from System.Collections.Generic import List
9  from Autodesk.Revit.DB import FilteredElementCollector, BuiltInCategory, BuiltInParameter
10
11 # DECLARAÇÃO DE VARIÁVEL DOCUMENTO ATIVO
12 doc = __revit__.ActiveUIDocument.Document
13
14 # COLETA AMBIENTES DENTRO DO DOCUMENTO ATIVO
15 rooms = FilteredElementCollector(doc).OfCategory(BuiltInCategory.OST_Rooms)
16
17 # ATRIBUIÇÃO DE LISTA areaMinima COM CÓDIGO DE OBRAS - RIBEIRÃO PRETO
18 areaMinima = [['Salas', 8.0], ['Quarto', 8.0], ['Copa', 3.0], ['Cozinha', 4.6], ['Banheiro', 2.5], ['Lavanderia', 2.5]]
19
20 # FAZ UM LOOP - NUMERO DE VEZES = N. AMBIENTES
21 for r in rooms:
22     # VERIFICA SE O AMBIENTE EXISTE
23     if r.Area > 0:
24
25         # COLETA OS DADOS DE CADA AMBIENTE
26         Area_ambiente = round ((r.Area * 0.3048)/3.2808, 2)
27         nome = Element.Name.GetValue(r)
28         nomeIniciais = nome [0:3]
29
30         # SELECIONA E VERIFICA A AREA DE CADA AMBIENTE EM FUNÇÃO DAS PRIMEIRAS 3 LETRAS
31         if nomeIniciais == 'Sal':
32
33             if Area_ambiente > areaMinima [0] [1]:
34
35                 print (str (nome) + " em M2 = " + str(Area_ambiente) + " => " + "Ambiente aprovado")
36
37             else:
38
39                 print (str (nome) + " em M2 = " + str(Area_ambiente) + " => " + "Ambiente REPROVADO")
40
41         if nomeIniciais == 'Qua':
42
43         if nomeIniciais == 'Cop':
44
45         if nomeIniciais == 'Coz':
46
47         if nomeIniciais == 'WC ':
48
49         if nomeIniciais == 'Lav':
50
51     print ("Rotina finalizada")
52

```

Python file length: 3,016 lines: 91

Fonte: Sena (2019).

Figura 26 – Resultado da execução da rotina criada por Sena (2019).



Fonte: Sena (2019).

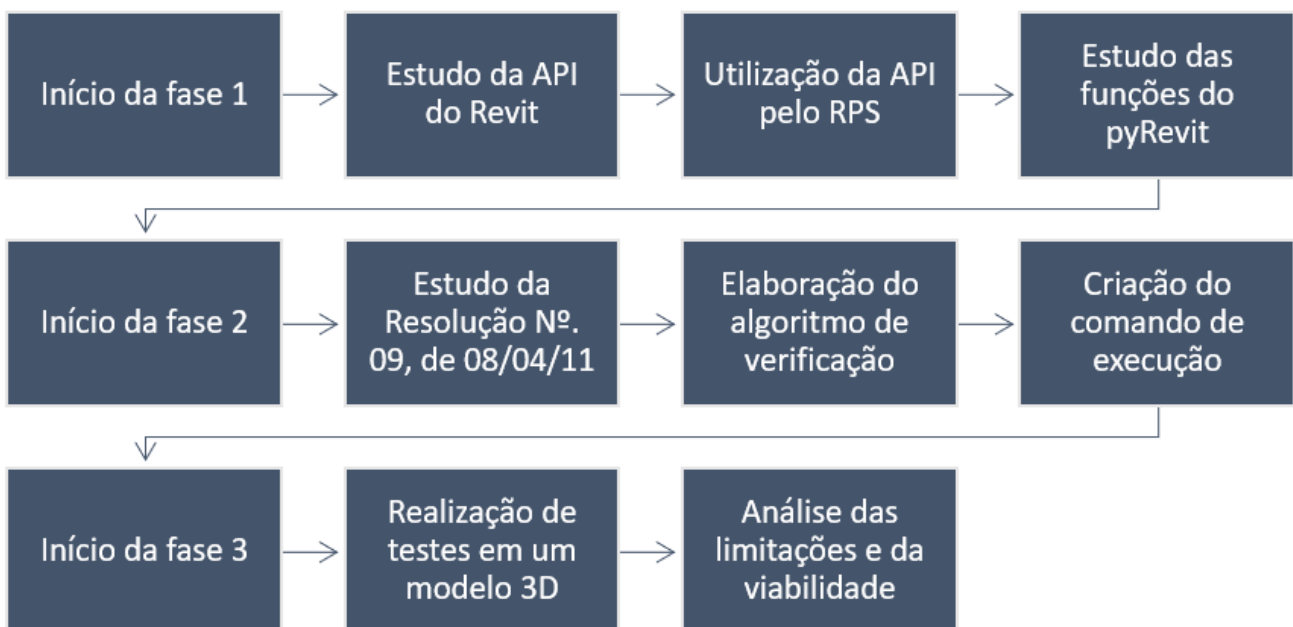
Como pode ser observado pela figura acima, o código criado por Sena (2019) gera uma janela com os dados referentes as áreas dos cômodos, é então efetuada uma comparação com os parâmetros da lei municipal de Ribeirão Preto e o resultado da comparação é imprimido nessa mesma janela, demonstrando a aprovação ou reprovação do ambiente em questão.

Capítulo 3 – Metodologia

A metodologia deste trabalho envolverá a realização de processos que terão como foco os objetivos específicos já listados. Com isso será possível associar a linguagem de programação Python com a *API* do *software* Autodesk Revit, permitindo a customização do programa, de forma a automatizar as verificações dos parâmetros estabelecidos pela Resolução N°. 09, de 08 de Abril de 2011, da ADASA.

Com o objetivo de facilitar a visualização e a organização das etapas em que este trabalho será executado, foi elaborado um fluxograma que divide a metodologia em 3 fases, servindo como um resumo e um passo a passo dos processos a serem realizados. O fluxograma pode ser observado na Figura 27 e as fases serão explicadas em mais detalhes posteriormente.

Figura 27 – Fluxograma com as fases deste trabalho.



Fonte: autor.

3.1. Primeira fase

A primeira etapa se inicia com o estudo da documentação disponibilizada pela Autodesk (AUTODESK, ©2021c), com a finalidade de se obter domínio sobre as funções e parâmetros que podem ser acessados por meio da *API* do Revit.

O acesso à *API* requer uma interface de programação, assim, será escolhida a linguagem Python para a elaboração dos códigos. A escolha dessa linguagem é devido a sua natureza dinâmica e de sintaxe simples, sendo a linguagem de programação ideal para se trabalhar com o objetivo de tratar e analisar dados, o que a torna fundamental para a associação com um modelo, que se baseia na metodologia *BIM* e, portanto, todos seus elementos são regidos por regras e parâmetros.

O *plugin Revit Python Shell* será utilizado, já que adiciona um console que implementa o *IronPython*, uma versão da linguagem Python que irá permitir o acesso e utilização da *API* disponibilizada pelo Revit. Esse *plugin* será empregado para a realização de testes e estudos da aplicação da linguagem Python associada ao Revit, já que proporciona um console onde é possível escrever códigos que interagem com o modelo e dão resultados quase que em tempo real.

Após ter o código devidamente testado e funcional, o *plugin pyRevit* será utilizado para implementar a solução no Revit. Com esse *plugin* será possível criar uma nova aba no Revit, contendo todos os novos comandos criados e que serão executados diretamente no *software* com apenas um clique. O *pyRevit* não necessita que o código seja inserido em um console para, então, ser executado, como ocorre no *Revit Python Shell*, sendo assim uma ótima opção para códigos já testados.

3.2. Segunda Fase

O algoritmo que será elaborado em Python e aplicado por meio dos *plugins* mencionados, tem como finalidade ser uma verificação automática para as diretrizes estabelecidas pela Resolução N°. 09, de 08 de Abril de 2011, definida pela ADASA. Assim, será

criado um comando que utilizará os dados do modelo para realizar os cálculos e as verificações estabelecidas pela Resolução.

A Resolução N°. 09, de 08 de Abril de 2011, da ADASA tem o seguinte objetivo: “estabelecer as diretrizes e critérios gerais para requerimento e obtenção de outorga de lançamento de águas pluviais em corpos hídricos superficiais do Distrito Federal e naqueles delegados pela União e estados.”. O documento traz diretrizes para limites e cálculo da vazão de lançamento de águas pluviais em corpo hídrico superficial, fórmulas para cálculo dos reservatórios, regras para redução da área impermeabilizada, entre outros.

O código a ser escrito em Python terá como foco a implementação dessas diretrizes, fazendo uma comparação com os valores presentes no modelo e os pré-definidos pela resolução. Posteriormente será analisado quais são as limitações do algoritmo elaborado e quais pré-requisitos devem ser apresentados pelo modelo para que o código seja executado com sucesso.

3.3. Terceira Fase

Em relação ao modelo 3D, não haverá a necessidade de modelá-lo do começo, já que este será cedido por um arquiteto e será utilizado na realização dos testes do algoritmo. Esse modelo possui um terreno com área maior que 600 m², se enquadrando nos critérios da Resolução N°. 09, de 08 de Abril de 2011.

Primeiramente serão realizados os cálculos manuais com os dados do modelo 3D de testes, seguindo as diretrizes da resolução. Após a obtenção dos resultados pelo cálculo manual, o programa desenvolvido será executado e os valores resultantes serão comparados com os alcançados inicialmente.

Por fim, após a aquisição de conhecimentos para um uso satisfatório da *API* do Revit associada a uma interface de programação com a linguagem Python, será avaliada a viabilidade da utilização dessas ferramentas, com base nas dificuldades encontradas no decorrer do trabalho e no processo de adaptação do modelo ao código criado.

Capítulo 4 – Análises e Resultados

4.1. Desenvolvimento de aplicações para o Autodesk Revit

A Autodesk disponibiliza para *download* em seu *website* um *Software Development Kit (SDK)* para o Revit (AUTODESK, ©2021c). Esse *SDK* inclui a documentação de referência para *API* do Revit, juntamente com exemplos de como utilizá-la. Um ponto a se destacar é que o material está disponível apenas em inglês.

Devido à complexidade do Autodesk Revit e considerando que é possível gerenciar quase que qualquer etapa de uma construção por ele, a funcionalidade da *API* acaba sendo bem extensa, possibilitando lidar com os aspectos arquitetônico, estrutural, mecânico, elétrico, hidráulico, com planejamento e orçamento e até mesmo com a parte de análise energética de um empreendimento. Portanto, para realizar o desenvolvimento de aplicações para o Revit, a consulta constante ao material de referência se faz imprescindível (Figura 28).

Figura 28 – Exemplo de uma página do documento de referência da API do Revit.

Revit 2021.1 API

Ocultar Localizar Voltar Encaminhar Parar Atualizar Início Imprimir Opções

Sumário Índice Pesquisar

BuiltInFailures.RebarFailures C
 BuiltInFailures.RebarShapeFai
 BuiltInFailures.RebarSystemFa
 BuiltInFailures.RefPlaneFailure
 BuiltInFailures.RegenFailures C
 BuiltInFailures.RenderFailures
 BuiltInFailures.RevisionCloudF
 BuiltInFailures.RevisionFailures
 BuiltInFailures.RevolutionFailur
 BuiltInFailures.RoofFailures Cla
 BuiltInFailures.RoomFailures C
 BuiltInFailures.ScheduleViewF
 BuiltInFailures.SculptingFailure
 BuiltInFailures.SectionViewFail
 BuiltInFailures.SelectionFailure
 BuiltInFailures.SheetFailures C
 BuiltInFailures.SiteFailures Cla
 BuiltInFailures.SiteImportFailure
 BuiltInFailures.SketchFailures C
 BuiltInFailures.SkylightFailures
 BuiltInFailures.SlabFailures Cla
 BuiltInFailures.SlabShapeFailu
 BuiltInFailures.SlantedColumnf
 BuiltInFailures.SpanDirectionS
 BuiltInFailures.SplineFailures C
 BuiltInFailures.StairRampFailur
 BuiltInFailures.SteelElementFa
 BuiltInFailures.SteelElementsFi
 BuiltInFailures.StructuralAssetf
 BuiltInFailures.StructuralConne
 BuiltInFailures.StructuralCustor
 BuiltInFailures.StructuralLoadF
 BuiltInFailures.SweepFailures C
 BuiltInFailures.SweptBlendFail
 BuiltInFailures.SystemNavigatc
 BuiltInFailures.SystemsFailures
 BuiltInFailures.TagFailures Cla
 BuiltInFailures.TextFailures Cla
 BuiltInFailures.TilePattern TileF
 BuiltInFailures.ToggleViewFail
 BuiltInFailures.TrussFailures Cl
 BuiltInFailures.UniqueElementI
 BuiltInFailures.ViewFailures Cla
 BuiltInFailures.WallFailures Cla
 BuiltInFailures.WallJoinFailures
 BuiltInFailures.WallJoinRoofFa
 BuiltInFailures.WorksharingFai

Collapse All Code: All

Revit 2021.1 API

Category Class

[Members](#) [Example](#) [See Also](#) [Send Feedback](#)

Represents the category or subcategory to which an element belongs.

Namespace: [Autodesk.Revit.DB](#)
Assembly: RevitAPI (in RevitAPI.dll) Version: 21.0.0.0 (21.1.1.109)

Syntax

C#

```
public class Category : APIObject
```

Visual Basic

```
Public Class Category _  
    Inherits APIObject
```

Visual C++

```
public ref class Category : public APIObject
```

Remarks

Categories are an import tool within Revit for identifying the inferred type of an element, such as anything in the Walls category should be considered as a wall. The API exposes access to the built in categories within Revit via the Document.Settings.Categories property.

Examples

C# [Copy](#)

```
Element selectedElement = null;  
foreach (ElementId id in uidoc.Selection.GetElementIds(  
{  
    selectedElement = document.GetElement(id);  
    break; // just get one selected element  
})  
  
// Get the category instance from the Category property  
Category category = selectedElement.Category;  
  
BuiltInCategory enumCategory = (BuiltInCategory)category
```

Fonte: autor.

Como pode ser observado pela parte lateral esquerda na Figura 28, há uma imensa quantidade de páginas no documento, referenciando classes, métodos e exemplos. Assim, o ponto principal para desenvolver as aplicações se torna aprender a como pesquisar no material de referência e entender como utilizar a classe ou o método em questão.

Outra questão bastante importante sobre os documentos de referência da *API* é que, como o próprio nome menciona, são referências e não guias. Portanto, o primeiro contato com o material e como utilizá-lo pode ser de difícil entendimento. Para esse caso a Autodesk também disponibiliza em seu *website* um guia para desenvolvedores utilizarem a *API* do Revit, chamado de *Revit API Developers Guide* (AUTODESK, ©2021c). O guia possui diversos exemplos, passo-a-passos e recomendações, sendo uma ótima opção para o primeiro contato com o desenvolvimento para o Revit.

De acordo com a Autodesk (©2021c), a *API* é uma *API* feita na plataforma .NET, então é possível utilizar diversas bibliotecas e utilidades presentes na plataforma .NET para realizar o desenvolvimento em conjunto com a *API* do Revit. Outro ponto a ser mencionado é que no material de referência e no guia para desenvolvedores, a principal linguagem de programação utilizada é a linguagem C# (C Sharp), conhecida pela sua utilização na plataforma .NET.

Por fim, ainda que a experiência com a plataforma .NET e com a linguagem C# não seja totalmente necessária para o desenvolvimento de aplicações para o Revit, é inegável que esses conhecimentos são extremamente úteis e quase que um pré-requisito para a utilização eficaz do guia de desenvolvedores e dos documentos de referência da *API*.

4.2. Processo de desenvolvimento do programa proposto

O processo de desenvolvimento na linguagem Python do programa que propõe a aplicação da Resolução N°. 09, de 08 de Abril de 2011 da ADASA no *software* Autodesk

Revit, pode ser dividido em três etapas principais, que serão abordadas neste tópico. As etapas são:

1. Utilização da linguagem Python para operações com a *API*.
2. Desenvolvimento de uma interface gráfica de usuário para o programa.
3. Implementação dos cálculos referentes à Resolução N° 09.

4.2.1. Python na utilização da *API* do Revit

Considerando que todos os materiais de referência e guias oficiais sobre a utilização da *API* tem como base a linguagem C#, a falta de conhecimento em como interpretar um código escrito nessa linguagem, traz dificuldades no aprendizado da *API* do Revit. A utilização de Python, a princípio, acaba sendo dependente desses conhecimentos, já que, utilizando apenas o material oficial da Autodesk, seria necessário interpretar os exemplos em C# e então aplicá-los na implementação da linguagem Python para a plataforma .NET, o *IronPython*.

Todavia, graças a esforços coletivos na criação e desenvolvimento dos projetos de código aberto *Revit Python Shell (RPS)* e *pyRevit*, existem formas de contornar, até um certo ponto, a necessidade de se ter familiaridade com a linguagem C# e com a plataforma .NET para desenvolver aplicações em Python para o Revit. As páginas dos repositórios de ambos os projetos no GitHub (THOMAS, [s.d]; EHSAN IRAN-NEJAD, [s.d.]) contam com uma série de referências e exemplos que auxiliam no aprendizado da utilização da linguagem Python com a *API* do Revit.

Em especial, o *pyRevit* conta com um *website* (PYREVIT, ©2021) onde há uma série de documentações sobre como o programa funciona, suas funcionalidades e bastante conteúdo, tanto em texto quanto em vídeo, sobre como desenvolver aplicações para o Revit com Python. Além disso, também conta uma pequena comunidade e fóruns em que é possível discutir ideias e solucionar dúvidas sobre o assunto. O projeto do *pyRevit* também conta com sua própria *API*, que funciona como um conjunto de funções que visam facilitar o uso da *API* do Revit com a linguagem Python (PYREVIT, ©2021).

Devido à falta de familiaridade com a linguagem C# e com a plataforma .NET, o conteúdo disponível no *website* do projeto do *pyRevit*, principalmente os vídeos voltados a iniciantes no desenvolvimento de aplicações para o Revit, foi essencial para o desenvolvimento deste trabalho. O *pyRevit* também traz a possibilidade de acessar o código dos programas que acompanham sua instalação. O estudo desses programas (Figura 29), que também foram escritos em Python, foi outro ponto indispensável para entender o processo de extrair dados e interagir, de um modo geral, com o modelo por meio da *API*.

Figura 29 – Trecho de código de um dos programas incluídos na instalação do *pyRevit*.

```
script.py X
s > Victor > AppData > Roaming > pyRevit-Master > extensions > pyRevitTools.extension > pyRevit.tab > Selection.pane
55 """Activates selection tool that picks a specific type of element.
54
53 Shift-Click:
52 Pick favorites from all available categories
51 """
50 # pylint: disable=E0401,W0703,C0103
49 from collections import namedtuple
48
47 from pyrevit import revit, UI
46 from pyrevit import forms
45 from pyrevit import script
44
43 # import this script's configurator
42 import pick_config
41
40
39 logger = script.get_logger()
38 my_config = script.get_config()
37
36
35 CategoryOption = namedtuple('CategoryOption', ['name', 'revit_cat'])
34
33
32 class PickByCategorySelectionFilter(UI.Selection.ISelectionFilter):
31     """Selection filter implementation"""
30     def __init__(self, category_opt):
29         self.category_opt = category_opt
28
27     def AllowElement(self, element):
26         """Is element allowed to be selected?"""
25         if element.Category \
24             and self.category_opt.revit_cat.Id == element.Category.Id:
23             return True
22         else:
21             return False
20
19     def AllowReference(self, refer, point): # pylint: disable=W0613
18         """Not used for selection"""
17         return False
16
```

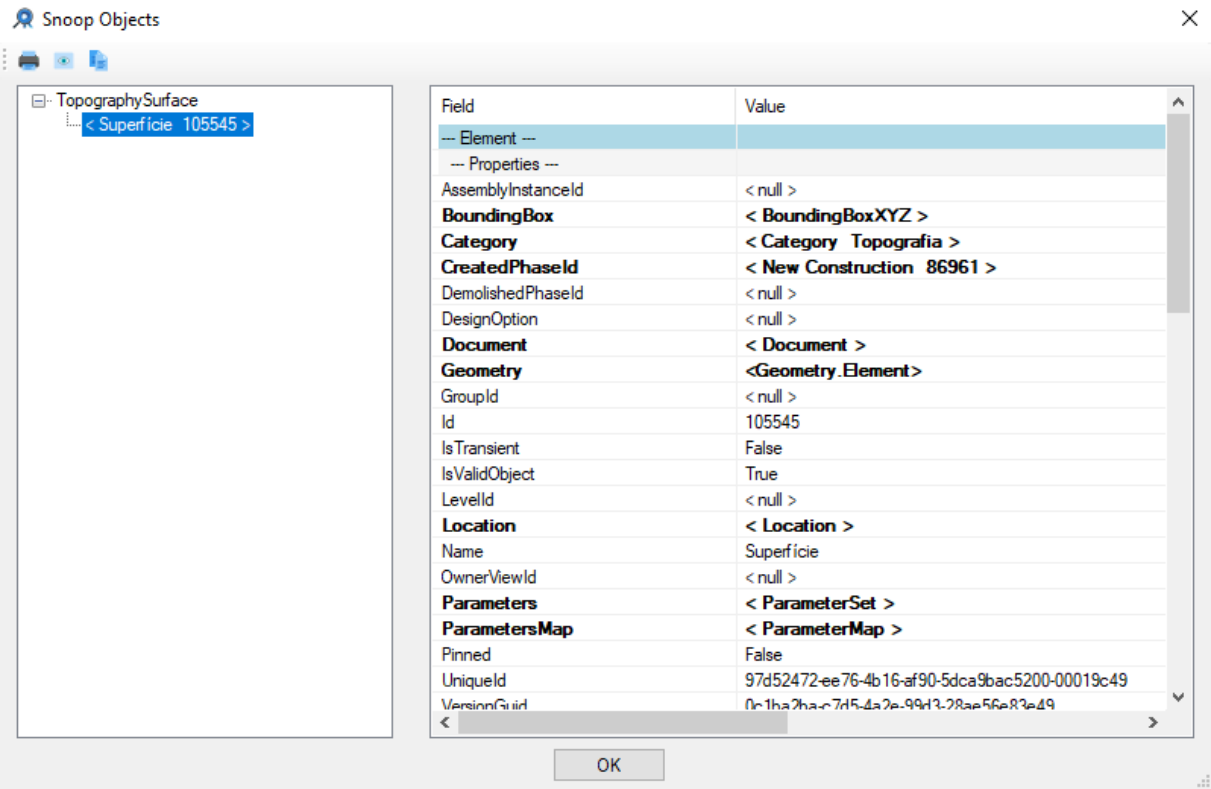
Fonte: autor.

Outra extensão que, apesar de não estar diretamente relacionada com a utilização da linguagem Python, foi extremamente valiosa foi o *RevitLookup*, uma ferramenta de

exploração da base de dados do projeto, que permite acessar os parâmetros, propriedades e relações de um elemento BIM (TAMMIK, [s.d]). Essa extensão auxiliou na identificação dos parâmetros que poderiam ser acessados em um determinado elemento e, posteriormente, na extração desses dados por meio do código em Python (Figura 30).

Sem o uso dessa extensão seria necessário fazer uma pesquisa na documentação da *API* do Revit para encontrar esses parâmetros e esse processo poderia demandar bastante tempo, já que a documentação é extensa e complexa. Já com o *RevitLookup*, esse processo se torna bem mais simples e rápido, pois é possível selecionar um elemento e checar diretamente todos os parâmetros relacionados a ele.

Figura 30 – Exemplo do uso do *RevitLookup* em um elemento de um modelo Revit.

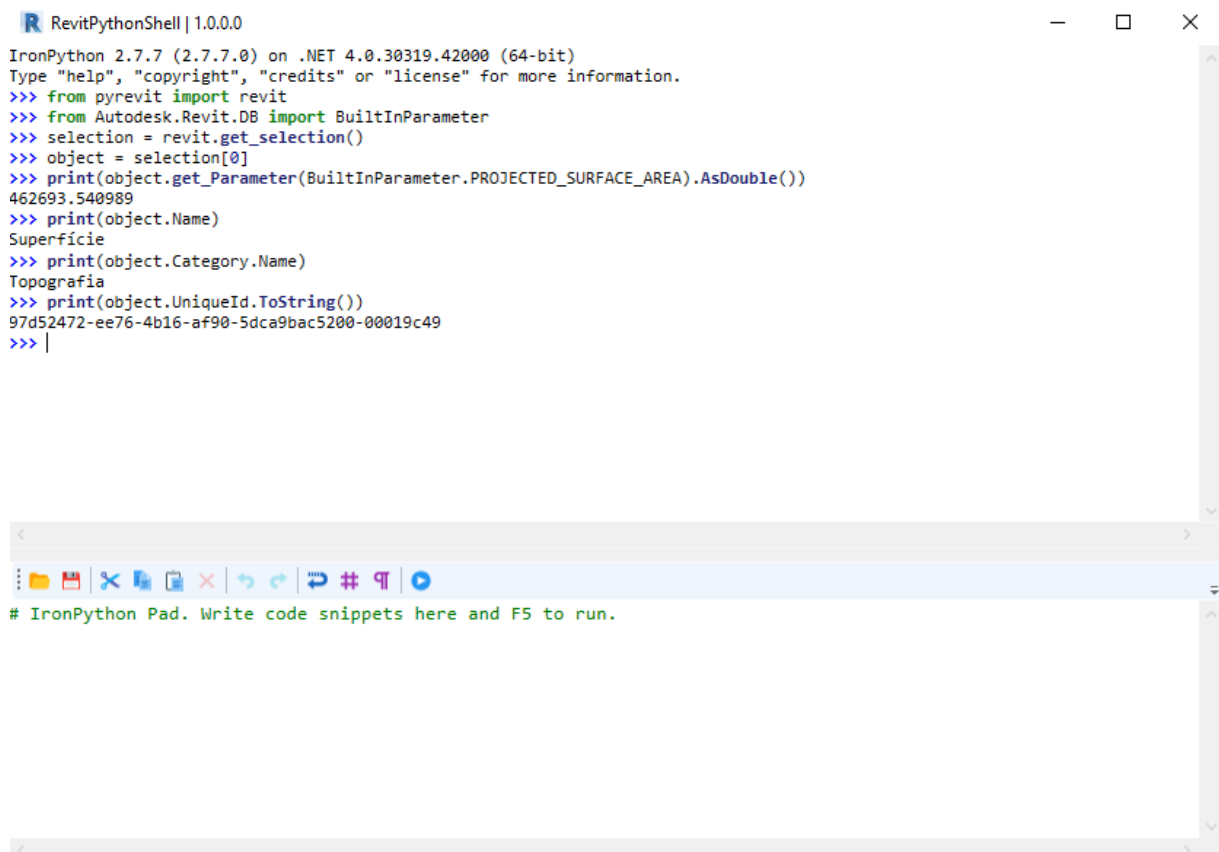


Fonte: autor.

4.2.2. Criação da interface gráfica do usuário

Com o uso do *Revit Python Shell* e com um pouco de conhecimento no uso de Python para acessar a *API* do Revit, já é possível interagir com um modelo (Figura 31), extrair dados e automatizar algumas tarefas. Esse é um ótimo método para realizar testes e aprender a como utilizar a *API* de uma forma interativa e dinâmica, já que dispensa o uso de um programa externo para editar e compilar o código, como é no caso da linguagem C#.

Figura 31 – Exemplo do uso do *Revit Python Shell* para extrair dados de um elemento.



```
RevitPythonShell | 1.0.0.0
IronPython 2.7.7 (2.7.7.0) on .NET 4.0.30319.42000 (64-bit)
Type "help", "copyright", "credits" or "license" for more information.
>>> from pyrevit import revit
>>> from Autodesk.Revit.DB import BuiltInParameter
>>> selection = revit.get_selection()
>>> object = selection[0]
>>> print(object.get_Parameter(BuiltInParameter.PROJECTED_SURFACE_AREA).AsDouble())
462693.540989
>>> print(object.Name)
Superfície
>>> print(object.Category.Name)
Topografia
>>> print(object.UniqueId.ToString())
97d52472-ee76-4b16-af90-5dca9bac5200-00019c49
>>> |

# IronPython Pad. Write code snippets here and F5 to run.
```

Fonte: autor.

Ainda que seja possível criar todo um programa por meio do *RPS* ou até mesmo rodar códigos já prontos, o usuário fica preso a ter sempre que abrir o *RPS* antes de

poder realizar qualquer tarefa, além de ser apresentado à uma interface não tão amigável, onde os dados são imprimidos em um console (como pode ser observado na Figura 31).

Portanto, para implementar um programa e disponibilizá-lo para um usuário comum, é interessante que haja uma interface gráfica que seja amigável e intuitiva, com janelas e botões, deixando o código em segundo plano. Caso o contrário, o primeiro contato com o programa pode gerar uma certa resistência por parte de alguns usuários. Um exemplo simples pode ser verificado na Figura 32, que imprime os mesmos dados da Figura 31, porém de uma forma mais amigável e deixando o código por trás dos panos.

Figura 32 – Exemplo de uma simples interface gráfica para o usuário.



Area:	462693.540989
Name:	Superfície
Category Name:	Topografia
Unique Id:	97d52472-ee76-4b16-af90-5dca9bac

Fonte: autor.

A API do Revit não possui ferramentas para a criação de uma interface gráfica, todavia, como já mencionado, também é possível fazer uso das bibliotecas presentes na plataforma .NET. Assim é possível utilizar o *Windows Presentation Foundation (WPF)*, uma estrutura de interface do usuário para criar aplicativos cliente de desktop utilizando o *Extensible Application Markup Language (XAML)* (MICROSOFT, 2021a).

De acordo com a Microsoft (2021b), o XAML é uma linguagem de marcação para programação de aplicações declarativas. Essa linguagem simplifica a criação de uma interface de usuário para aplicativos .NET, permitindo um fluxo de trabalho onde é

possível separar a criação da interface da lógica do aplicativo e usando ferramentas diferentes (MICROSOFT, 2021c).

O estudo do *WPF* e do *XAML* foi essencial para a criação da interface gráfica deste projeto e também foi a etapa que mais demandou tempo e esforços. Ainda que existam mais exemplos e uma documentação mais rica (disponibilizada pela Microsoft) quando comparado com a *API* do Revit, existe um problema em comum quando o assunto é desenvolvimento com Python: a grande maioria dos exemplos e dos materiais estão associados à linguagem C#.

Após o estudo do funcionamento básico da linguagem *XAML*, o desafio nessa etapa do projeto foi como criar a lógica da aplicação utilizando o *WPF* juntamente com Python e, posteriormente, associar o programa com a *API* do Revit para interagir com o modelo. Novamente, o material no *website* do projeto *pyRevit* (PYREVIT, ©2021), juntamente com o acesso ao código dos programas que vem em sua instalação, foi imprescindível para o progresso no aprendizado dessas tecnologias.

4.2.3. Implementação dos cálculos referentes à Resolução Nº 09

Com a interface gráfica do usuário pronta e com o conhecimento em como interagir com o modelo por meio da *API* e exibir dados na interface, a implementação dos cálculos referentes à Resolução Nº 09 de 08 de Abril de 2011 se torna relativamente simples e foi a parte que demandou menos tempo para ser desenvolvida neste trabalho.

A implementação da resolução consistiu em entender quais tipos de dados deveriam ser extraídos do modelo (como a área e o tipo do elemento), aplicar as equações descritas no tópico 2.5 deste trabalho e, por fim, disponibilizar os resultados para o usuário.

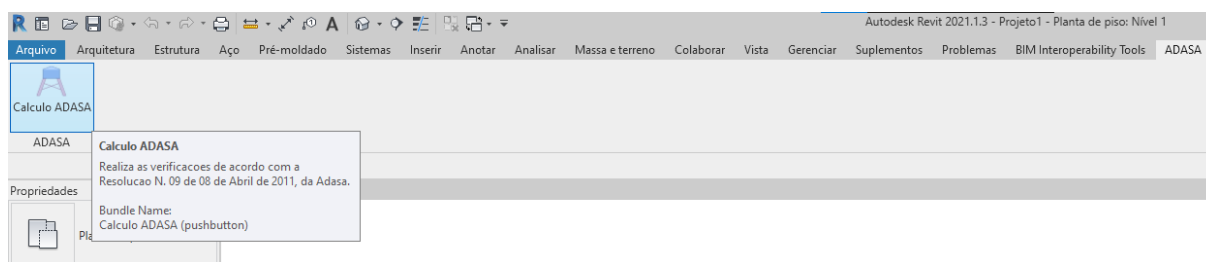
Ainda que a implementação dos cálculos tenha sido feita com sucesso, o programa exige uma certa sensibilidade por parte do usuário para entrar com alguns dados necessários para os cálculos e para selecionar os elementos a serem considerados.

Esses pontos serão discutidos em mais detalhes a seguir, no tópico referente ao funcionamento do programa desenvolvido.

4.3. Utilização do programa desenvolvido

A princípio deve ser mencionado que, como o programa foi desenvolvido com a utilização da *API* do *pyRevit*, a instalação prévia deste se faz necessária. O *pyRevit* também permite que se crie uma nova aba no Revit e sejam adicionados diversos programas nessas abas em formato de botões. A aplicação desenvolvida neste projeto pode ser acessada dessa forma, como pode ser visto na Figura 33.

Figura 33 – Botão de acesso ao programa desenvolvido, presente na aba “ADASA”.



Fonte: autor.

Após clicar no botão “Cálculo ADASA” será gerada a interface gráfica criada para este projeto (Figura 34). Essa interface será o contato inicial que o usuário terá com o programa.

Figura 34 – Interface do programa desenvolvido.

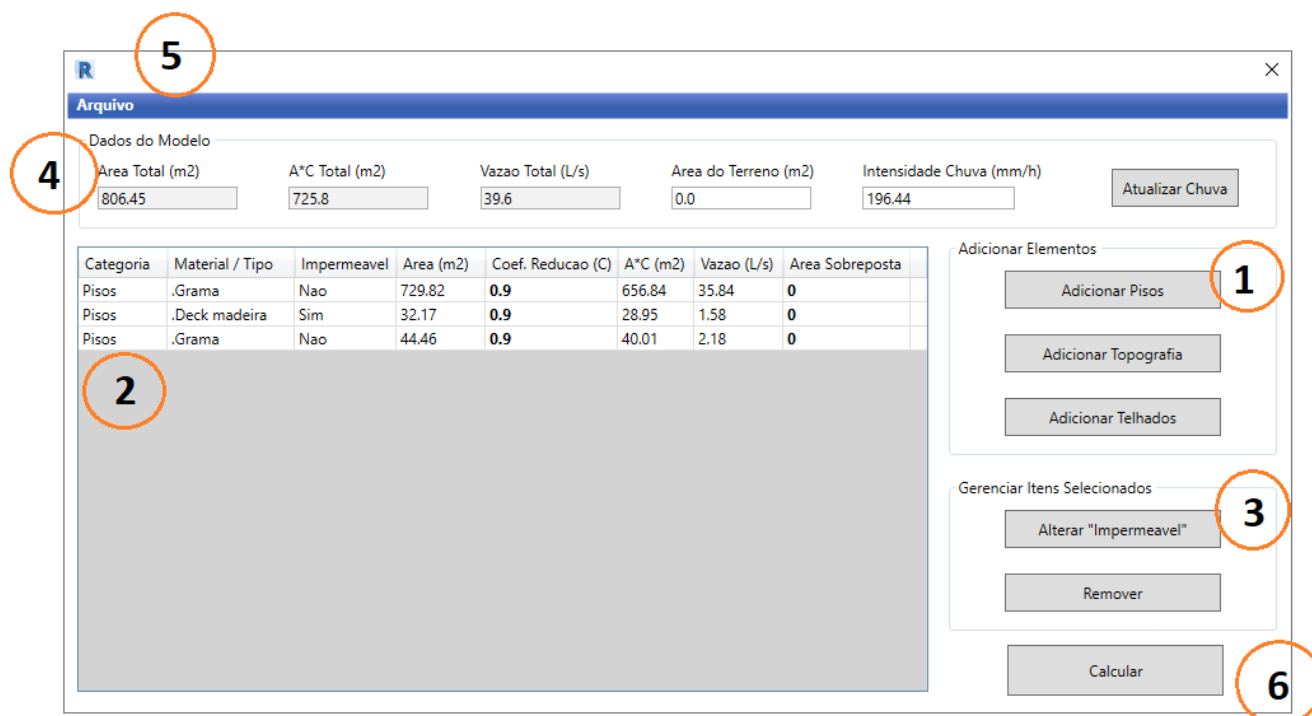
The interface is titled 'Arquivo' and contains the following sections:

- Dados do Modelo:** A group of input fields and a button.
 - Area Total (m2): 0
 - A°C Total (m2): 0
 - Vazao Total (L/s): 0
 - Area do Terreno (m2): 0.0
 - Intensidade Chuva (mm/h): 196.44
 - Atualizar Chuva button
- Table:** A table with 8 columns: Categoria, Material / Tipo, Impermeavel, Area (m2), Coef. Reducao (C), A°C (m2), Vazao (L/s), and Area Sobreposta. The table body is currently empty.
- Adicionar Elementos:** A section with three buttons: Adicionar Pisos, Adicionar Topografia, and Adicionar Telhados.
- Gerenciar Itens Selecionados:** A section with two buttons: Alterar "Impermeavel" and Remover.
- Calcular:** A large button at the bottom right.

Fonte: autor.

Em relação a utilização efetiva do programa, há alguns pontos a serem mencionados. A Figura 35 traz a mesma interface após a adição de alguns elementos de um modelo teste. Nessa figura também estão marcados alguns pontos para facilitar a explicação do uso do programa.

Figura 35 – Pontos importantes da interface gráfica da aplicação.



Fonte: autor.

O entendimento dos pontos da Figura 35 é essencial para uma boa utilização do programa. A explicação de cada ponto se dá da seguinte forma:

- 1. Bloco "Adicionar Elementos":** nesse bloco há três botões para efetuar a adição dos elementos que serão considerados no cálculo. Ao clicar em "Adicionar Pisos", por exemplo, o usuário voltará ao modelo e, ao selecionar elementos, apenas elementos do tipo "Piso" serão considerados na seleção. Os outros dois botões funcionam de forma análoga.
- 2. Lista de elementos selecionados:** após selecionar os elementos, estes aparecerão nessa lista e algumas informações serão automaticamente extraídas ou calculadas do modelo. O usuário deve estar atento ao "Coef. Redução (C)", o valor padrão que será proposto pelo programa é de 0.9, porém esse valor pode ser alterado ao clicar na célula correspondente. Outra célula que pode ser alterada é a célula "Area Sobreposta" que deve ser

utilizada quando o usuário optar por desconsiderar parte da área do objeto selecionado.

3. **Bloco “Gerenciar Itens Selecionados”**: esse bloco conta com dois botões que alteram as propriedades dos itens dentro da lista de elementos que foram selecionados pelo usuário, sendo possível remover da lista ou alterar a propriedade “Impermeável” de um elemento, caso seja necessário.
4. **Bloco “Dados do Modelo”**: o somatório dos valores apresentados na lista de elementos será apresentado aqui. O usuário deve entrar com a área total do terreno a ser considerada para os cálculos. O valor de intensidade da chuva padrão que o programa traz é de 196.44 mm/h, porém o usuário pode alterá-lo conforme sua necessidade. Caso haja alguma alteração na intensidade da chuva, o botão “Alterar Chuva” deve ser utilizado para efetuar as devidas modificações nos cálculos.
5. **Menu “Arquivo”**: esse botão traz duas opções: salvar e carregar dados (em formato CSV). Assim o usuário tem a possibilidade de salvar determinada configuração de elementos selecionados e checar seus dados ou até mesmo utilizá-la posteriormente com a opção de carregar dados.
6. **Botão “Calcular”**: após a adição de todos os elementos que devem ser considerados no cálculo, seus devidos coeficientes de redução e a área total do terreno, pode-se fazer uso do botão “Calcular” que gerará uma nova janela, computando os resultados de acordo com a Resolução N° 09.

A janela que será gerada pelo botão “Calcular” é a janela de resultados que trará as seguintes informações: vazão total e de pré-desenvolvimento em L/s (litros por segundo), m³/h (metros cúbicos por hora) e m³/s (metros cúbicos por segundo), se há necessidade de reservatório para a área considerada e, por fim, o volume do reservatório em m³ (metros cúbicos), considerando um reservatório de quantidade. Na Figura 36 pode ser observada a interface da janela de resultados.

Figura 36 – Interface da janela de resultados.

The image shows a software window titled 'Arquivo' with a blue header bar. It contains two sections for flow data and a section for reservoir information. The first section, 'Vazao Total', has three input fields for L/s (270.53), m3/h (973.91), and m3/s (0.27053). The second section, 'Vazao de Pre-Desenvolvimento', has three input fields for L/s (13.44), m3/h (48.38), and m3/s (0.01344). Below these is a checkbox for 'Necessidade de Reservatorio' which is checked, with options 'Sim' and 'Nao'. The 'Volume do Reservatorio (m3)' is shown as 233.26. At the bottom are two buttons: 'Voltar' and 'Salvar Resultados'.

Vazao Total		
L/s	m3/h	m3/s
270.53	973.91	0.27053

Vazao de Pre-Desenvolvimento		
L/s	m3/h	m3/s
13.44	48.38	0.01344

Necessidade de Reservatorio: ☒ Sim ☐ Nao

Volume do Reservatorio (m3): 233.26

Voltar Salvar Resultados

Fonte: autor.

4.4. Viabilidade e limitações

A ideia inicial ao fim da segunda fase era ter um botão que, ao ser acionado, executaria o programa e imprimiria os resultados sem a necessidade de o usuário interagir com o programa. Todavia, essa abordagem exigiria a criação de um programa muito complexo devido à complexidade e a liberdade criativa que o próprio Autodesk Revit dispõe para os seus usuários.

Em outras palavras, o Revit permite a criação de um modelo de diversas formas diferentes, com nomenclaturas, objetos, materiais, parâmetros e famílias que podem ser criados e customizados pelo usuário, assim, escrever um código que leva todos esses detalhes em consideração seria complexo e criaria um programa propenso a erros.

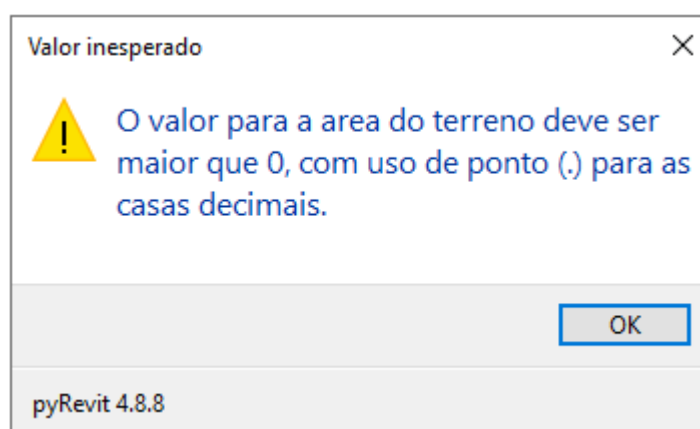
Portanto, o programa desenvolvido não executa os cálculos automaticamente ao ser acionado, mas apresenta uma etapa intermediária que exige a interação por parte do

usuário. Assim, o usuário deve selecionar os elementos a serem levados em consideração no modelo 3D e pode alterar alguns coeficientes que serão utilizados no cálculo. Dessa forma, há a necessidade de um certo conhecimento, por parte do usuário, da Resolução N°. 09, de 08 de abril de 2011, da ADASA, para que o programa seja utilizado com eficácia.

Dessa maneira, os erros de cálculo que poderão surgir na tela de resultados estarão atrelados, de uma forma geral, aos dados inseridos pelo usuário. Consequentemente, deve-se ter um maior cuidado ao selecionar os elementos a serem considerados, verificando se não há sobreposições entre eles, por exemplo: pisos que estão em pavimentos diferentes, um acima do outro, provavelmente não devem ser selecionados juntos. O usuário também deve se atentar aos valores de coeficiente de redução e intensidade da chuva, já que são dados fundamentais para os cálculos.

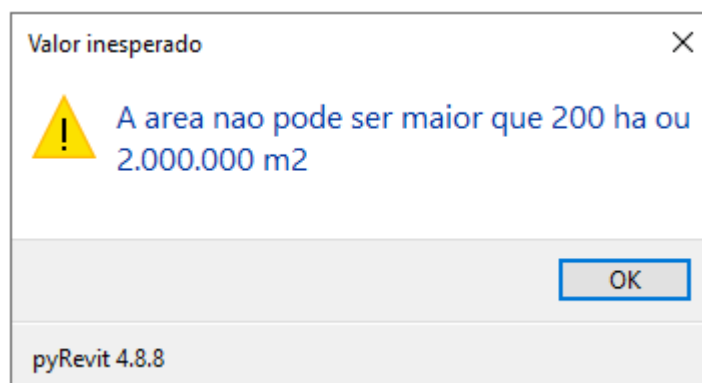
Por fim, o valor correspondente à área total do terreno também deve ser inserido corretamente. Para o cálculo de reservatórios de quantidade, a Resolução N°. 09, de 08 de abril de 2011, da ADASA, em seu capítulo 4 afirma que a equação para determinar o volume (utilizada neste projeto) só é válida para uma área de contribuição inferior a 200 ha (duzentos hectares). Assim, caso haja a entrada de valores incompatíveis, o programa irá apresentar mensagens de erro (Figuras 37 e 38).

Figura 37 – Mensagem de erro para valores incompatíveis de área do terreno.



Fonte: autor.

Figura 38 – Mensagem de erro para terrenos acima de 200 ha.



Fonte: autor.

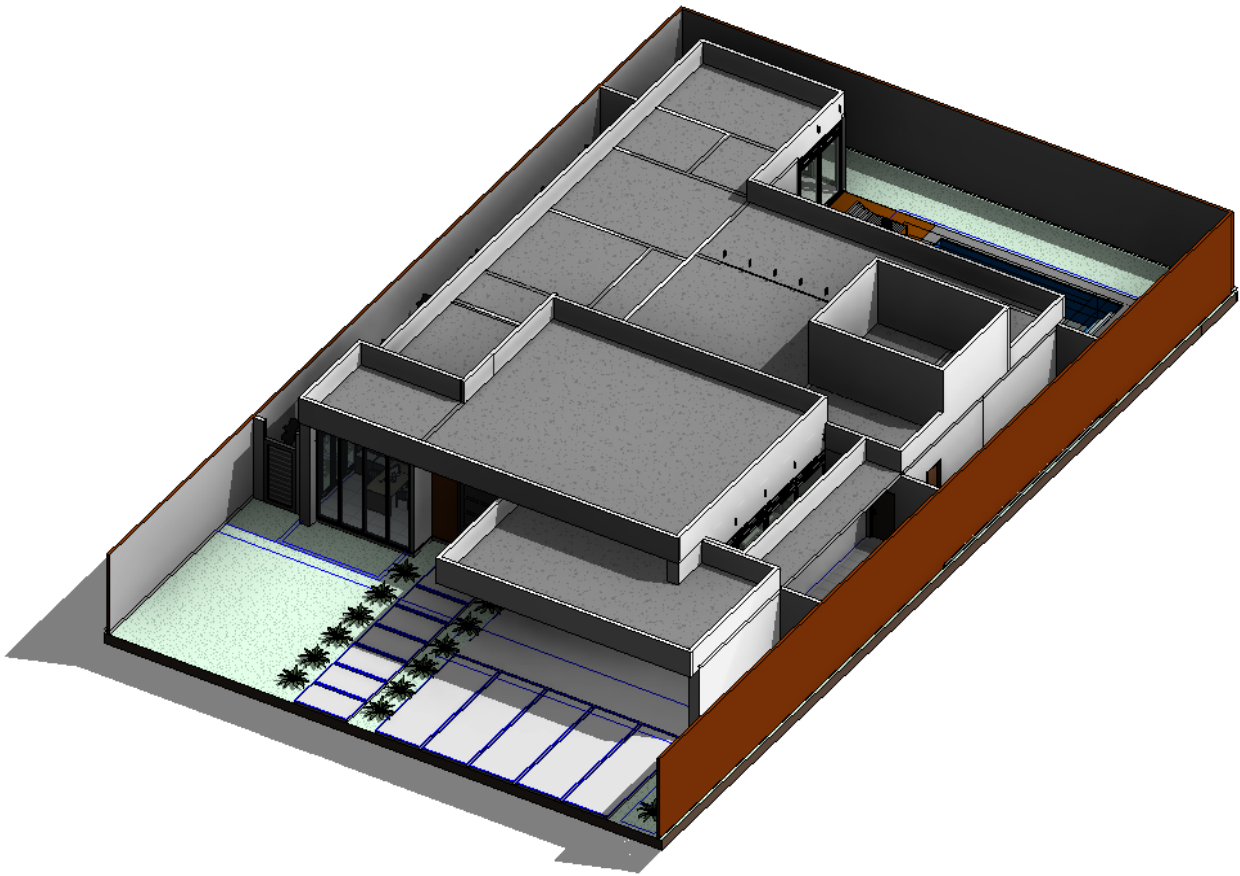
O artigo 5º da resolução afirma que o limite da vazão específica de 24,4 L/(s.ha) (litro por segundo por hectare), relacionado ao lançamento de águas pluviais em corpo hídrico superficial, pode ser desconsiderado para terrenos com área inferior a 600 m² e destinados a habitação unifamiliar. O programa desenvolvido permite a utilização de áreas inferiores a 600 m², já que não há verificações se a habitação é unifamiliar ou não.

4.5. Testes e resultados

4.5.1. Primeiro teste: modelo 3D cedido por um arquiteto

Um modelo 3D foi cedido por um arquiteto para a realização de testes do programa desenvolvido. O modelo é uma habitação unifamiliar com área de terreno de 757,56 m² e pode ser observado a seguir nas Figuras 39 e 40.

Figura 39 – Vista 3D do modelo cedido para testes.



Fonte : autor.

Figura 40 – Planta baixa do modelo cedido.



Fonte: autor.

Com o objetivo de demonstrar mais a fundo como os cálculos são realizados e, portanto, reforçar a credibilidade do programa, os cálculos para o modelo de testes serão primeiramente feitos manualmente. O processo de cálculo será semelhante ao que é executado pelo programa e as referências para as equações e valores utilizados podem ser encontradas no tópico 2.5 deste trabalho e na Resolução Nº 09 de 08 de Abril de 2011, da ADASA.

O primeiro passo é o cálculo da intensidade da chuva utilizando os valores de tempo de recorrência de 10 anos e tempo de concentração de 5 minutos, conforme a Resolução Nº 09.

$$i = \frac{21,7 \times F^{0,16}}{(tc + 11)^{0,815}} = \frac{21,7 \times 10^{0,16}}{(5 + 11)^{0,815}} = 3,274 \frac{mm}{min} = 196,44 \frac{mm}{h} \quad (6)$$

Após a determinação da intensidade da chuva é necessário dividir o modelo para contabilizar as áreas impermeáveis e então essas áreas serão multiplicadas por um coeficiente de escoamento superficial. De acordo com a Tabela 1, o valor de 0,9 será utilizado para todas as áreas deste modelo. Por fim, utiliza-se da equação do método racional para se obter a vazão de cada área. Esses processos estão resumidos na Tabela 2, a seguir.

Tabela 2 – Levantamento manual das áreas impermeáveis do modelo de testes.

Levantamento das áreas impermeáveis					
Descrição	Área (m2)	Coeficiente C	A x C	Intensidade da chuva (mm/h)	Vazão (L/s)
Laje impermeabilizada	382,19	0,9	343,97	196,44	18,77
Pisos área externa (piscina)	76,87	0,9	69,18		3,78
Pisos entrada	61,2	0,9	55,08		3,01
Piso cerâmica	14,1	0,9	12,69		0,69
TOTAL	534,36		480,92		26,24

Fonte: autor.

O valor total da vazão deve ser utilizado para efetuar a comparação com a vazão de pré-desenvolvimento. Caso o valor da vazão calculado na Tabela 2 seja maior que a vazão de pré-desenvolvimento, haverá a necessidade da implantação de um reservatório. De acordo com a Resolução Nº 09 a vazão de pré-desenvolvimento pode ser calculada da seguinte forma:

$$Q = 24,4 \frac{L}{s \times ha} \times \frac{757,56}{10.000} ha = 1,85 L/s \quad (7)$$

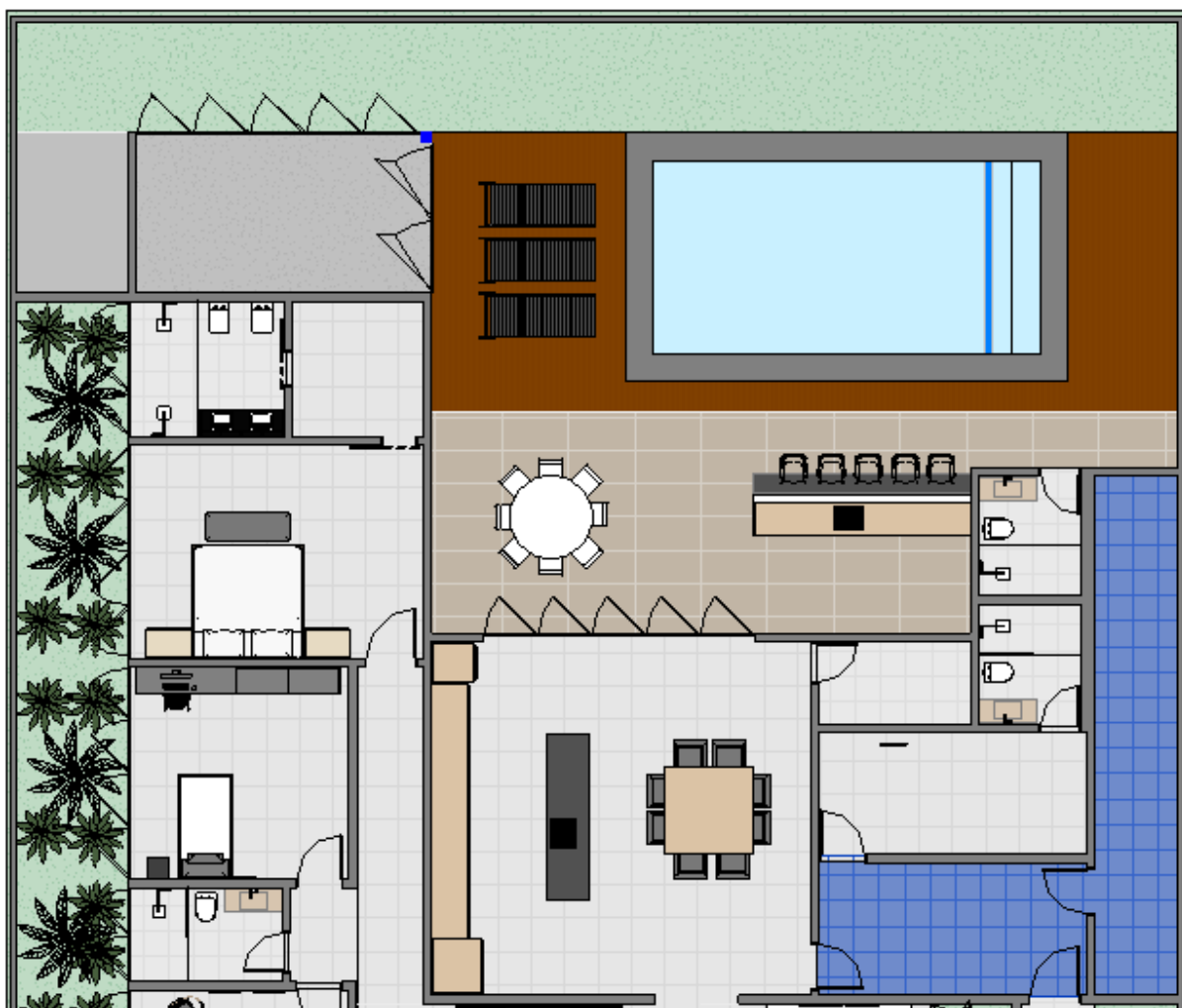
Como pode ser verificado pelo resultado da equação acima, o uso do reservatório é necessário. O cálculo do seu volume é feito utilizando a área total do terreno em hectares e o percentual da área impermeável, calculado pela divisão do total A*C da Tabela 2 pela área total de contribuição. O resultado é o seguinte:

$$V = 4,705 \times \left(\frac{480,92}{757,56} \times 100 \right) \times 0,075756 = 22,62 m^3 \quad (8)$$

O resultado do cálculo manual foi a necessidade de um reservatório de 22,62 m³ (metros cúbicos). O processo será refeito, agora utilizando o programa desenvolvido neste trabalho. As etapas para a melhor utilização do programa estarão descritas a seguir.

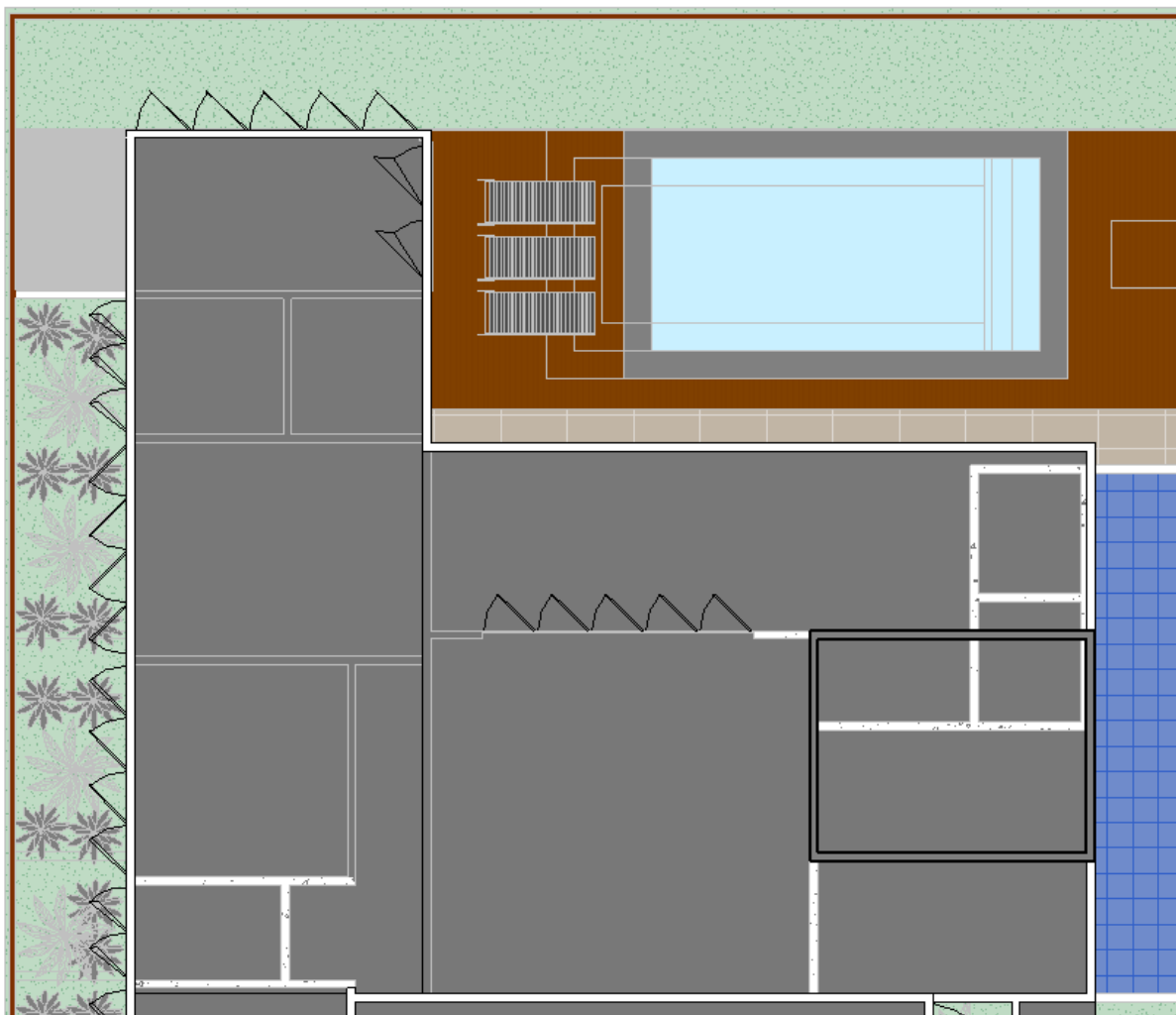
O primeiro passo para a utilização correta do programa é identificar as áreas que devem ser consideradas para o cálculo e seus tipos (topografia, piso ou telhado). No modelo de testes não há telhados e nem topografias, portanto adicionaremos apenas pisos ao utilizar o programa. Outro ponto que o usuário deve se atentar é a presença de áreas sobrepostas, estas devem ser identificadas e devidamente desconsideradas. As Figuras 41 e 42 ilustram um exemplo de área sobreposta presente no modelo de testes.

Figura 41 – Parte da área destaca em azul está sobreposta pela laje da edificação.



Fonte: autor.

Figura 42 – Área da Figura 41 com a vista da cobertura.



Fonte: autor.

Após a verificação das áreas sobrepostas, caso elas existam, há basicamente duas formas para se lidar com essa situação: a primeira opção é dividir o objeto em dois objetos diferentes no modelo 3D e a segunda opção é verificar a área que está sobreposta e entrar com esse valor na célula “Área Sobreposta” presente no programa.

O próximo passo, após identificar as áreas que devem ser consideradas e verificar a presença de áreas sobrepostas, é utilizar o programa para fazer o levantamento das áreas impermeáveis. Também deve-se utilizar alguma referência (como o Termo de Referência e Especificações para Elaboração de Projetos de Sistema de Drenagem Pluvial

no Distrito Federal, publicado em 2019 pela NOVACAP) para verificar o coeficiente de escoamento superficial (célula “Coef. Reducao (C)” no programa).

A Figura 43 demonstra o levantamento das áreas impermeáveis do modelo cedido para testes e a Figura 44 demonstra o resultado calculado pelo programa.

Figura 43 – Levantamento das áreas impermeáveis do modelo de testes.

Arquivo

Dados do Modelo

Area Total (m2)

534.36

A°C Total (m2)

480.92

Vazao Total (L/s)

26.25

Area do Terreno (m2)

757.56

Intensidade Chuva (mm/h)

196.44

Atualizar Chuva

Categoria	Material / Tipo	Impermeavel	Area (m2)	Coef. Reducao (C)	A°C (m2)	Vazao (L/s)	Area Sobreposta
Pisos	15cm - LAJE IMPE	Sim	50.82	0.9	36.97	2.02	9.74
Pisos	15cm - LAJE IMPE	Sim	120.89	0.9	108.8	5.94	0
Pisos	15cm - LAJE IMPE	Sim	83.46	0.9	75.11	4.1	0
Pisos	15cm - LAJE IMPE	Sim	21.39	0.9	19.25	1.05	0
Pisos	15cm - LAJE IMPE	Sim	105.31	0.9	94.78	5.17	0
Pisos	15cm - LAJE IMPE	Sim	10.06	0.9	9.05	0.49	0
Pisos	15cm - Cerâmica	Sim	28.3	0.9	12.69	0.69	14.2
Pisos	porcelanato bege	Sim	42.75	0.9	7.83	0.43	34.05
Pisos	.Deck madeira	Sim	32.17	0.9	28.95	1.58	0
Pisos	15cm - Pastilhas	Sim	21	0.9	18.9	1.03	0
Pisos	15cm - Pastilhas	Sim	1.75	0.9	1.58	0.09	0
Pisos	15cm - Pastilhas	Sim	1.75	0.9	1.58	0.09	0
Pisos	5cm Granito	Sim	11.5	0.9	10.35	0.56	0
Pisos	10cm - Concreto	Sim	61.2	0.9	55.08	3.01	0

Adicionar Elementos

Adicionar Pisos

Adicionar Topografia

Adicionar Telhados

Gerenciar Itens Selecionados

Alterar "Impermeavel"

Remover

Calcular

Fonte: autor.

Figura 44 – Resultado da execução do programa no modelo de testes.

Vazao Total		
L/s	m3/h	m3/s
26.25	94.5	0.02625

Vazao de Pre-Desenvolvimento		
L/s	m3/h	m3/s
1.85	6.66	0.00185

Necessidade de Reservatorio: ☒ Sim ☐ Nao

Volume do Reservatorio (m3): 22.63

Voltar Salvar Resultados

Fonte: autor.

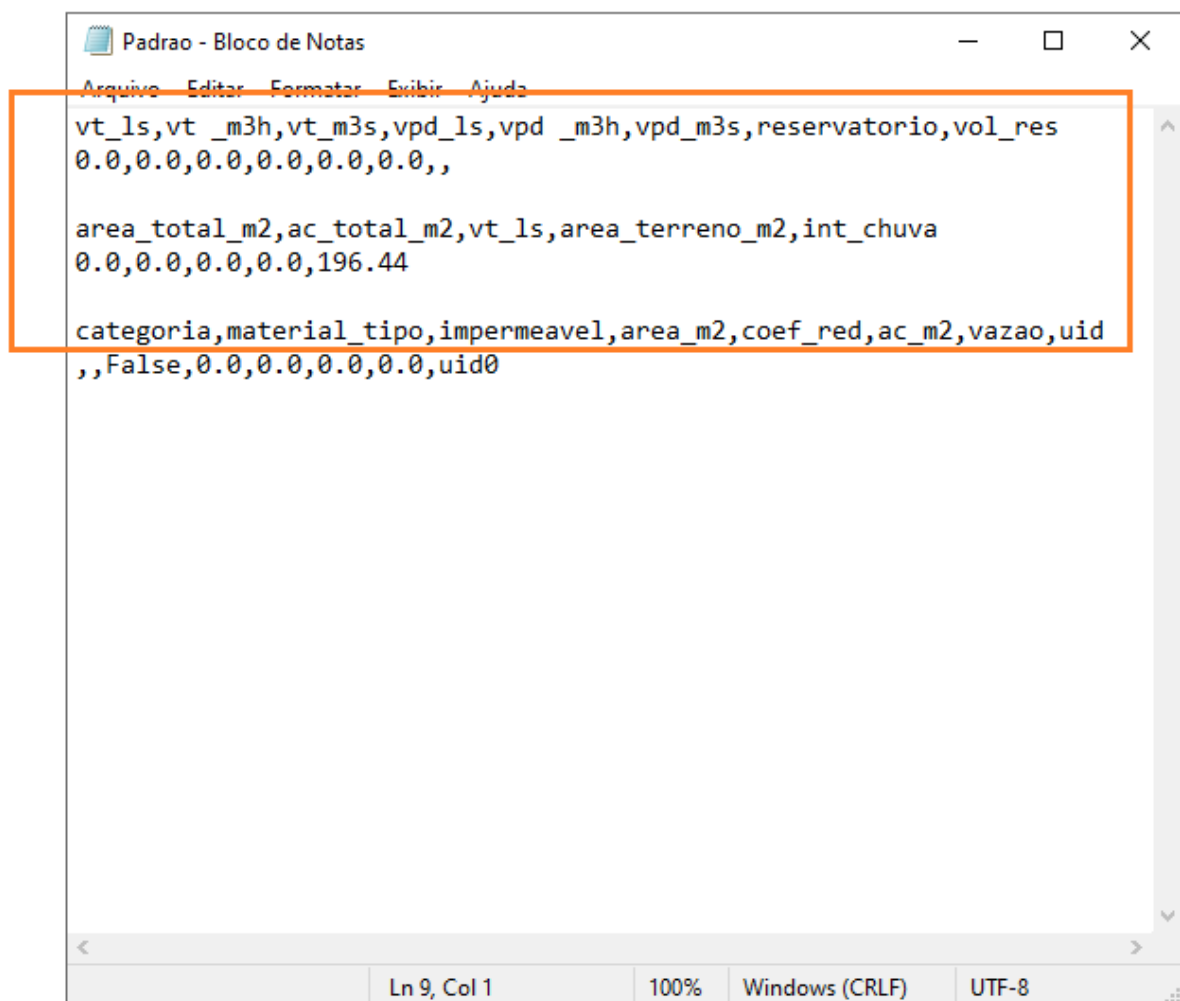
Comparando os valores presentes na Figura 44 com os valores calculados manualmente é possível observar que os resultados do programa estão de acordo com os obtidos manualmente, com diferenças mínimas devido aos arredondamentos.

4.5.2. Segundo teste: dimensionamento realizado por Braga (2018)

Braga (2018) efetuou em seu trabalho o dimensionamento do reservatório de retenção de águas pluviais de uma edificação de 1200 m² (metros quadrados) utilizando as diretrizes da Resolução Nº 09 de 08 de Abril de 2011, da ADASA. O segundo teste consistirá em utilizar os dados desse empreendimento no programa e comparar os resultados com os obtidos por Braga (2018).

Como não é possível ter acesso a um modelo 3D da edificação que possa ser acessado pelo Autodesk Revit, o uso do programa se dará de uma forma alternativa. É possível montar um arquivo em formato *CSV (Comma Separated Values)* com os dados a serem utilizados que será carregado no programa e, então, os cálculos poderão ser efetuados. O padrão de como deve ser montado o arquivo será demonstrado na Figura 45.

Figura 45 – Padrão de arquivo que pode ser carregado no programa desenvolvido.



```
vt_ls,vt_m3h,vt_m3s,vpd_ls,vpd_m3h,vpd_m3s,reservatorio,vol_res
0.0,0.0,0.0,0.0,0.0,0.0,,

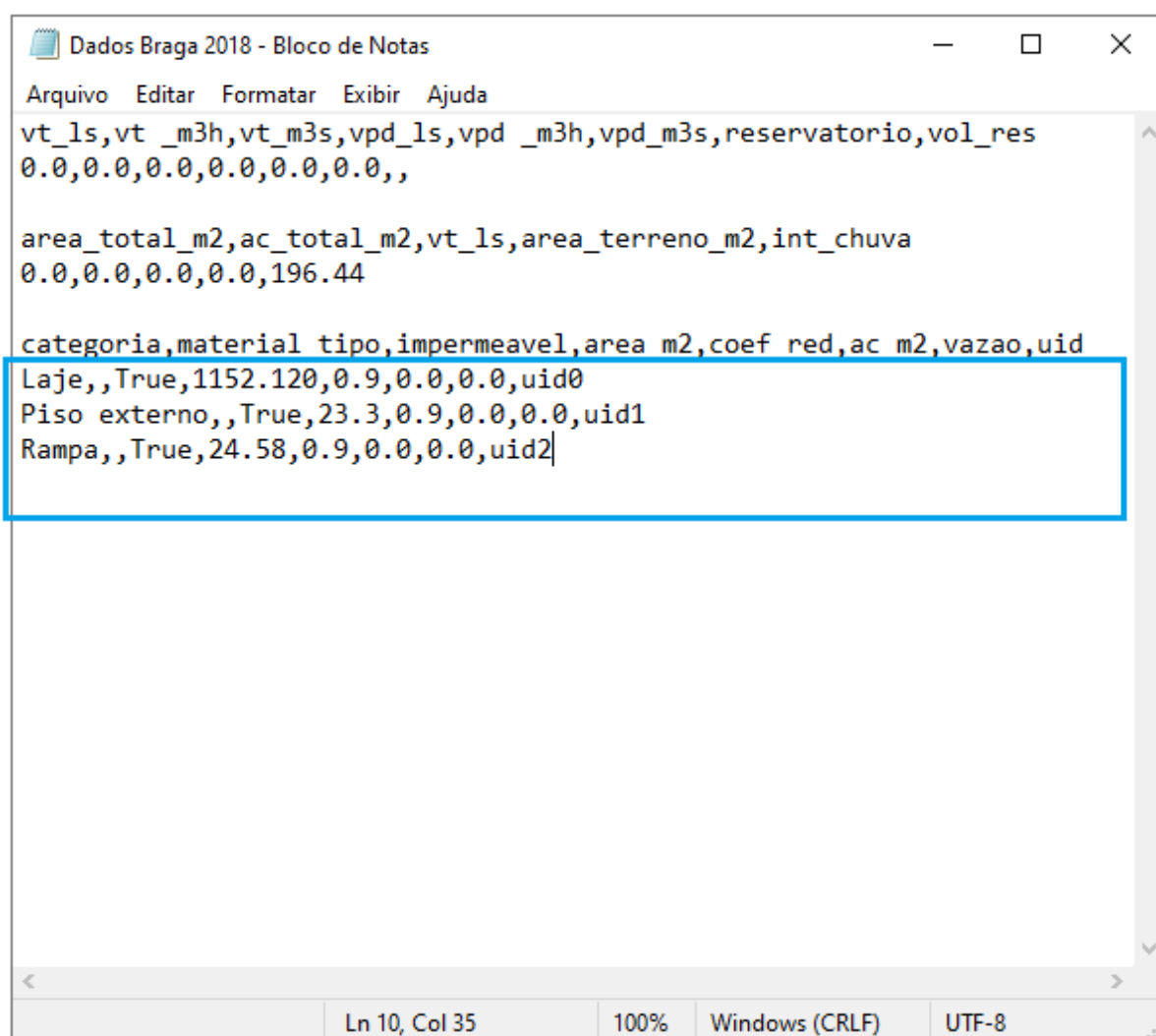
area_total_m2,ac_total_m2,vt_ls,area_terreno_m2,int_chuva
0.0,0.0,0.0,0.0,196.44

categoria,material_tipo,impermeavel,area_m2,coef_red,ac_m2,vazao,uid
,,False,0.0,0.0,0.0,0.0,uid0
```

Fonte: autor.

O retângulo alaranjado na Figura 45 representa valores que não devem ser alterados para que o programa leia corretamente os dados. A linha abaixo do retângulo alaranjado representa um exemplo de dados de um objeto. Para adicionar um novo objeto basta adicionar novas linhas com a quantidade correta de dados (de acordo com a linha “categoria,material_tipo...” separados por vírgula. A Figura 46 irá ilustrar melhor como realizar esse procedimento, com os dados presentes no trabalho de Braga (2018).

Figura 46 – Arquivo CSV com os dados do trabalho de Braga (2018).



```
Dados Braga 2018 - Bloco de Notas
Arquivo  Editar  Formatar  Exibir  Ajuda
vt_ls,vt_m3h,vt_m3s,vpd_ls,vpd_m3h,vpd_m3s,reservatorio,vol_res
0.0,0.0,0.0,0.0,0.0,0.0,,
area_total_m2,ac_total_m2,vt_ls,area_terreno_m2,int_chuva
0.0,0.0,0.0,0.0,196.44
categoria,material_tipo,impermeavel,area_m2,coef_red,ac_m2,vazao,uid
Laje,,True,1152.120,0.9,0.0,0.0,uid0
Piso externo,,True,23.3,0.9,0.0,0.0,uid1
Rampa,,True,24.58,0.9,0.0,0.0,uid2
```

Fonte: autor.

O retângulo azul representa os dados que serão carregados no programa e foram editados manualmente. O primeiro valor é a categoria do objeto, o segundo valor é o material (como não há informações sobre o material, esse valor foi deixado em branco), o terceiro valor é se o material é impermeável ou não e os únicos valores aceitos são “True” ou “False”, o quarto valor é área do objeto, o quinto valor é o coeficiente de escoamento superficial e os demais valores não são necessários, porém devem estar presentes, conforme a Figura 46.

Abrindo o programa no Autodesk Revit, acessando o menu “Arquivo”, então clicando na opção “Carregar Dados” e, por fim, selecionando o arquivo CSV montado manualmente, caso não haja erros, o programa carregará os dados e será possível obter os resultados normalmente (Figuras 47 e 48).

Figura 47 –Interface do programa após carregar os dados de Braga (2018).

The screenshot shows a software window titled 'Arquivo' with a blue header. Below the header, there is a section 'Dados do Modelo' with five input fields: 'Area Total (m2)' (1200), 'A*C Total (m2)' (1080), 'Vazao Total (L/s)' (58.93), 'Area do Terreno (m2)' (1200.00), and 'Intensidade Chuva (mm/h)' (196.44). To the right of these fields is a button labeled 'Atualizar Chuva'. Below this section is a table with the following data:

Categoria	Material / Tipo	Impermeavel	Area (m2)	Coef. Reducao (C)	A*C (m2)	Vazao (L/s)	Area Sobreposta
Rampa		Sim	24.58	0.9	22.12	1.21	0
Piso externc		Sim	23.3	0.9	20.97	1.14	0
Laje		Sim	1152.12	0.9	1036.91	56.58	0

To the right of the table, there is a section 'Adicionar Elementos' with three buttons: 'Adicionar Pisos', 'Adicionar Topografia', and 'Adicionar Telhados'. Below this is a section 'Gerenciar Itens Selecionados' with three buttons: 'Alterar "Impermeavel"', 'Remover', and 'Calcular'.

Fonte: autor.

Figura 48 – Resultado da execução do programa com os dados de Braga (2018).

The screenshot shows a software window titled 'R' with a standard Windows interface. It features a menu bar with 'Arquivo' and a close button. The main content area is divided into sections for flow rate calculations and reservoir volume. The 'Vazao Total' section displays three values: 58.93 L/s, 212.15 m3/h, and 0.05893 m3/s. The 'Vazao de Pre-Desenvolvimento' section displays three values: 2.93 L/s, 10.55 m3/h, and 0.00293 m3/s. Below these, there is a checkbox for 'Necessidade de Reservatorio' which is checked, with options 'Sim' and 'Nao'. The 'Volume do Reservatorio (m3)' is shown as 50.81. At the bottom, there are two buttons: 'Voltar' and 'Salvar Resultados'.

Vazao Total		
L/s	m3/h	m3/s
58.93	212.15	0.05893

Vazao de Pre-Desenvolvimento		
L/s	m3/h	m3/s
2.93	10.55	0.00293

Necessidade de Reservatorio: ☒ Sim ☐ Nao

Volume do Reservatorio (m3): 50.81

Voltar Salvar Resultados

Fonte: autor.

Os resultados obtidos pela execução do programa apresentaram os mesmos valores que os obtidos pelos cálculos manuais realizados por Braga (2018) em seu trabalho. O objetivo deste segundo teste foi demonstrar uma forma alternativa de utilizar o programa desenvolvido, salientando que é possível utilizá-lo mesmo sem a presença de um modelo aberto no Autodesk Revit.

Capítulo 5 – Conclusões

O desenvolvimento do programa proposto foi um sucesso, possibilitando alcançar os objetivos iniciais deste trabalho. O programa criado como plugin para o Autodesk Revit auxilia profissionais da construção civil na adequação de seus projetos às diretrizes da Resolução Nº 09 de 08 de abril de 2011 da ADASA. O plugin realiza os cálculos necessários de acordo com a resolução e automaticamente entrega ao usuário o volume que o reservatório deve ter para cada caso.

Mesmo que o programa já apresente vantagens aos profissionais que o utilizarem, já que permite economizar tempo e dinheiro ao automatizar os processos de cálculos presentes na resolução, há espaço para melhorias. Para isso um estudo mais aprofundado das funcionalidades do Autodesk Revit e de sua *API* se torna necessário.

Tratando somente da *API* do Revit, a utilização da linguagem Python por meio das extensões *Revit Python Shell* e *pyRevit*, permite a criação de aplicações de forma rápida e interativa, além de dispensar o uso de um compilador. Mas a predominância de exemplos e materiais na linguagem C# é um fator que dificulta o uso do Python.

Portanto, para criação de programas mais simples o uso de Python é viável. Já para se ter o domínio da *API* do Revit e utilizá-la de uma forma mais profunda e otimizada é necessário ter conhecimentos em C# e na plataforma .NET. Em todos os casos, a experiência com C# e com a plataforma .NET é um ponto que auxilia bastante no uso da *API*.

Ainda que o uso de Python não ser o ideal para o desenvolvimento de aplicações com a *API* do Revit, há um fator que não pode ser ignorado a respeito da linguagem: é uma das principais linguagens de programação utilizadas nos campos de *machine learning* e inteligência artificial. Essas áreas vêm crescendo cada vez mais nos últimos anos e se tornando parte fundamental de vários setores da indústria.

Assim, um contato maior com a linguagem por parte dos profissionais e acadêmicos da engenharia civil, pode vir a ser a porta de entrada para a pesquisa e a criação de

modelos de *machine learning* e inteligência artificial que auxiliem diretamente em toda a etapa de desenvolvimento e planejamento de projetos baseados na metodologia *BIM*. A incorporação dessas tecnologias pode ser um fator fundamental para uma mudança mais profunda na modernização do setor da construção civil.

Além disso, o desenvolvimento deste projeto exigiu constantes pesquisas a fim de entender e aprender a utilizar certos termos, tecnologias e linguagens (C# e XAML) que eram necessários para progredir na construção do código por trás da aplicação. Os problemas encontrados no decorrer do projeto que necessitaram de pesquisa para serem solucionados, revelaram que, com uma rápida procura na internet, era possível encontrar pessoas que tiveram problemas similares e outras pessoas que ajudaram a explicar o que possivelmente estava acontecendo e como resolver esses problemas.

Assim, além de existir uma gama de materiais e documentações gratuitos, um dos pontos mais fortes do setor da tecnologia e desenvolvimento é ter uma grande comunidade engajada e com espírito colaborativo. Projetos de código aberto como *Revit Python Shell*, *pyRevit* e até mesmo de grandes empresas de tecnologia são a prova de como essa comunidade é o ponto chave para o contínuo desenvolvimento da área.

Outro fator que vale a pena ser mencionado é como grandes empresas, universidades e até mesmo pessoas físicas criam bastante conteúdo gratuito e de qualidade. Ou seja, há um consenso no mundo da tecnologia que fomentar a disseminação de conhecimento de qualidade contribui para fortalecer essa comunidade e beneficia o setor como um todo.

Tendo esse contexto em vista, não é à toa que a área de tecnologia e desenvolvimento vem crescendo cada vez mais e cada vez mais rápido. As inovações continuam aparecendo em um ritmo espantoso e seus efeitos podendo ser percebidos nas mais diversas esferas da economia e da sociedade. Todo esse caráter dinâmico e de constante evolução tem um grande contraste com a indústria da construção que, por sua vez, principalmente no Brasil, é conhecida pela sua cultura enraizada em métodos construtivos arcaicos e por ter uma certa resistência à modernização.

Portanto, aproximar o engenheiro civil da habilidade técnica da programação não significa apenas trazer a possibilidade de criar aplicações que auxiliem em sua profissão, mas se configura em uma tentativa de incorporar, tanto para os acadêmicos quanto para os profissionais do setor, parte da cultura do mundo da tecnologia, tornando comum o uso da internet de forma mais engajada para criar uma comunidade forte e disseminar conhecimento, além de ter uma busca mais intensa pela evolução e modernização da área.

Referências

- ALRASHED, I.; KANTAMANENI, K. **A 5D Building Information Model (BIM) for Potencial Cost-Benefit Housing: A Case of Kingdom of Saudi Arabia (KSA)**. 2018. 15 f. *Infrastructures* 2018, 3, 13; doi:10.3390/infrastructures3020013. Disponível em <<https://www.mdpi.com/2412-3811/3/2/13/pdf-vor>>. Acesso em: 28 de jun. de 2021.
- ALVES, J. P. C. **Interoperabilidade BIM em projeto de Estruturas**. 2018. 92 f. Dissertação de Mestrado Integrado em Engenharia Civil – Faculdade de Ciências e Tecnologia, Universidade de Coimbra, Coimbra, 2018. Disponível em: <<https://estudogeral.sib.uc.pt/handle/10316/84911>>. Acesso em: 22 de jun. de 2021.
- ARAUJO, Edgar Oliveira. **Um Estudo de Automação Visando o Aumento de Produtividade na Construção Civil**. 2018. 102 f. Projeto de Graduação (Graduação em Engenharia Civil) – Escola Politécnica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2018.
- ASCENCIO, A. F. G.; CAMPOS, E. A. V. **Fundamentos da Programação de Computadores: algoritmos, PASCAL, C/C++ (padrão ANSI) e Java**. 3ª Edição. São Paulo: Pearson Education do Brasil, 2012. ISBN: 978-85-64574-16-8.
- AUTODESK. **Autodesk Developer Network**. ©2021a. Online. Disponível em: <<https://www.autodesk.com/developer-network/app-store>>. Acesso em: 19 de jul. de 2021.
- AUTODESK. **Revit: Multidisciplinary BIM software for higher-quality, coordinated designs**. ©2021b. Online. Disponível em: <<https://www.autodesk.com/products/revit/overview>>. Acesso em: 19 de jul. de 2021.
- AUTODESK. **Revit Developer Center**. ©2021c. Online. Disponível em: <<https://www.autodesk.com/developer-network/platform-technologies/revit>>. Acesso em: 19 de jul. de 2021.
- BAIA, D. V S. **Uso de ferramentas BIM para o planejamento de obras da construção civil**. 2015. Dissertação de Mestrado em Estruturas e Construção Civil, Publicação E. DM -

019A/15, Departamento de Engenharia Civil e Ambiental, Universidade de Brasília. Brasília-DF, 99 p.

BARBOSA, A. C. M. **A metodologia BIM 4D e BIM 5D aplicada a um caso prático: Construção de uma ETAR na Argélia**. 2014. 157 p. Dissertação (Mestrado em Engenharia Civil) – Instituto Superior de Engenharia do Porto. Porto, Portugal, 2014. Disponível em: <<https://recipp.ipp.pt/handle/10400.22/5571>>. Acesso em: 10 de jun. 2021.

BARRETO, E. M. M.; RIBEIRO, C. C.; ALMEIDA, M. L. B.; BAMBERG, P.; RIBEIRO, S. E. C.; OLIVEIRA, D. M. **Estudo da interoperabilidade de sistemas BIM por meio do formato IFC**. 2020. Brazilian Journal of Development, v.6, n.6, p.36059-36072, Curitiba, jun. 2020. ISSN 2525-8761. Disponível em <<https://www.brazilianjournals.com/index.php/BRJD/article/view/11444>>. Acesso em: 27 de jun. de 2021.

BIOTTO, C. N.; FORMOSO, C. T.; ISATTO, E. L. **O uso da modelagem BIM 4D no projeto e gestão de sistemas de produção em empreendimentos de construção**. ENTAC – XIV Encontro Nacional de Tecnologia do Ambiente Construído, Juiz de Fora, 2012.

BRAGA, W. M. S. **Dimensionamento do reservatório de retenção de águas pluviais da edificação do Tribunal Regional do Trabalho em Taguatinga-DF**. 2018. Trabalho de Conclusão de Curso (Graduação em Engenharia Civil) – Faculdade de Tecnologia e Ciências Sociais Aplicadas, Centro Universitário de Brasília, Brasília, 2018. Disponível em: <<https://repositorio.uniceub.br/jspui/handle/prefix/14015>>. Acesso em: 24 de jan. de 2022.

BRAICK, P. R.; MOTA, M. B. **História: das cavernas ao terceiro milênio**. 3ª edição, São Paulo: Moderna, 2007.

BRITO, D. M. de; FERREIRA, E. de A. M. **Avaliação de estratégias para representação e análise do planejamento e controle de obras utilizando modelos BIM 4D**. Ambiente Construído, Porto Alegre, v. 15, n. 4, p. 203-223, out./dez. 2015. ISSN 1678-8621 Associação Nacional de Tecnologia do Ambiente Construído. <http://dx.doi.org/10.1590/s1678-86212015000400047>

BUILDINGSMART. **Industry Foundation Classes (IFC) – An Introduction**. ©2021. Online. Disponível em: <<https://technical.buildingsmart.org/standards/ifc>>. Acesso em: 26 de jun. de 2021.

BUILDINGSMART. **What is openBIM?**. ©2021. Online. Disponível em: <<https://www.buildingsmart.org/about/openbim/openbim-definition/>>. Acesso em: 24 de jun. de 2021.

CAMPESTRINI, T. F.; GARRIDO, M. C.; MENDES, R.; SCHEER, S.; FREITAS, M. C. D. **Entendendo BIM: Uma visão do projeto de construção sob o foco da informação**. 2015. 1ª Edição. Curitiba, Paraná, 2015. Disponível em: <<https://www.campestrini.com.br/entendendobim>>. Acesso em: 27 de jun. de 2021.

CARNEIRO, N; MACIEL, A.C. **O uso da metodologia BIM 4D e BIM 5D para o gerenciamento de obras: Revisão Sistemática da Literatura**. In: ENCONTRO NACIONAL DE TECNOLOGIA DO AMBIENTE CONSTRUÍDO, 18., 2020, Porto Alegre. Anais... Porto Alegre: ANTAC, 2020.

CATELANI, W. S. **Implementação do BIM para construtoras e incorporadoras: Volume 1 - Fundamentos BIM**. Brasília, DF: CBIC - Câmara Brasileira da Indústria da Construção, 2016. 124 p. Disponível em: <https://cbic.org.br/inovacao/en_US/2017/10/18/coletanea-bim/>. Acesso em: 22 de jun. de 2021.

CATELANI, W. S. **Implementação do BIM para construtoras e incorporadoras: Volume 2 – Implementação BIM**. Brasília, DF: CBIC - Câmara Brasileira da Indústria da Construção, 2016. 72 p. Disponível em: <https://cbic.org.br/inovacao/en_US/2017/10/18/coletanea-bim/>. Acesso em: 22 de jun. de 2021.

CATELANI, W. S. **Implementação do BIM para construtoras e incorporadoras: Volume 3 – Colaboração e Integração BIM**. Brasília, DF: CBIC - Câmara Brasileira da Indústria da Construção, 2016. 132 p. Disponível em: <https://cbic.org.br/inovacao/en_US/2017/10/18/coletanea-bim/>. Acesso em: 22 de jun. de 2021.

CBIC – CÂMARA BRASILEIRA DA INDÚSTRIA DA CONSTRUÇÃO. **PIB Brasil e Construção Civil**. Online. Disponível em: <<http://www.cbicdados.com.br/menu/pib-e-investimento/pib-brasil-e-construcao-civil>>. Acesso em: 24 de jun. de 2021.

COELHO, P. M. N. **Rumo à Indústria 4.0**. 2016. 65 f. Dissertação de Mestrado em Engenharia e Gestão Industrial – Faculdade de Ciências e Tecnologia, Universidade de Coimbra, Coimbra, 2016. Disponível em: <<https://estudogeral.uc.pt/handle/10316/36992>>. Acesso em: 12 de jul. de 2021.

DEITEL, H. M.; DEITEL, P. J. **C++ Como Programar**. 5ª Edição. São Paulo: Pearson Prentice Hall, 2006. ISBN: 978-85-7605-056-8.

DEITEL, H.M.; DEITEL, P. J.; LIPERI, J.; WIEDERMANN, B. **Python How to Program**. 1ª ed. Pearson, 2002. ISBN: 978-0130923615,

DIAS, D. P. B.. **Planejamento em BIM 4D comparado ao planejamento tradicional**. Engineering Sciences, v.8, n.2, p.26-44, 2020. DOI: <http://doi.org/10.6008/CBPC2318-3055.2020.002.0004>

EASTMAN, C.; TEICHOLZ, P.; SACKS, R.; LISTON, K. **Manual de BIM: um guia de modelagem da construção para arquitetos, engenheiros, gerentes, construtores e incorporadores**. Tradução de Cervantes Gonçalves Ayres Filho et al. Porto Alegre, RS: BOOKMAN, 2014.

EHSAN IRAN-NEJAD. **pyRevit: Rapid Application Development (RAD) Environment for Autodesk Revit**. Online. Disponível em: <<https://github.com/eirannejad/pyRevit>>. Acesso em: 22 de jul. de 2021.

FARIAS, C. C. de O. **Aplicação da Tecnologia Building Information Modeling (BIM) no Planejamento de Obras 4D**. 2020. 52 f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Civil) – Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, Cajazeiras, Paraíba, 2020.

FERNANDES, J. P. M. **A metodologia Building Information Modeling aplicada ao projeto de estruturas**. 2013. 177 f. Dissertação (Mestrado). Escola de Engenharia, Universidade do Minho, Braga - Portugal. Disponível em: <<http://repositorium.sdum.uminho.pt/handle/1822/30793>>. Acesso em: 24 de jun. de 2021.

FERREIRA, B. F. V. **Aplicação de conceitos BIM à instrumentação de estruturas**. 2011. 128 f. Dissertação (Mestrado). Faculdade de Engenharia, Universidade do Porto, Porto -

Portugal. Disponível em: <<https://repositorio-aberto.up.pt/bitstream/10216/62046/1/000149954.pdf>>. Acesso em: 23 de jun. de 2021.

FREIRE, G. H. A. **Interoperabilidade no processo BIM utilizando Industry Foundation Classes (IFC) para Modelagem de Estruturas**. Rio de Janeiro, 2015. 141p. Dissertação de Mestrado — Departamento de Engenharia Civil, Pontifícia Universidade Católica do Rio de Janeiro.

HARRIS, J. **Integration of BIM and Business Strategy**. Masters of Project Management Program (Department of Civil and Environmental Engineering) - Northwestern University, Evanston, IL: McCormick School of Engineering and Applied Science, 2010. Disponível em: <<https://docplayer.net/12320707-Integration-of-bim-and-business-strategy-joe-harris.html>>. Acesso em: 10 de jun. 2021.

HERYANDI, A. **Developing Application Programming Interface (API) for Student Academic Activity Monitoring using Firebase Cloud Messaging (FCM)**. 2018. 7 f. IOP Conf. Series: Materials Science and Engineering 407 (2018) 012149. Disponível em: <<https://iopscience.iop.org/article/10.1088/1757-899X/407/1/012149>>. Acesso em: 29 de jun. de 2021.

JIN, B.; SAHNI, S.; SHEVAT, A. **Designing Web APIs: Building APIs That Developers Love**. 1ª ed. Sebastopol, California: O'Reilly Media, 2018. ISBN: 1492026921

KYMMELL, W. **Building Information Modeling: Planning and Managing Construction Projects with 4D CAD and Simulations**. Editora: McGraw-Hill, New York, 2008. 270 p. ISBN 9780071494533.

LAAKSO, M.; KIVINIEMI, A. **The IFC standard - a review of history, development, and standardization**. 2012. Journal of Information Technology in Construction (ITcon), Vol. 17, pg. 134 – 161. Disponível em: <<http://www.itcon.org/2012/9>>. Acesso em 22 de jun. 2021.

LABIB, R. **Is computer programming beneficial to architects and architecture students for complex modeling and informed performative design decisions?** Proceedings of the 12th Conference of Advanced Building Skins. Bern, Switzerland, 2017.

LIEBICH, T. *IFC 4: the new buildingSMART standard*. 2013. 25 f. *buildingSMART*, mar. 2013. Disponível em: <https://standards.buildingsmart.org/IFC/RELEASE/IFC4/buildingSMART_IFC4_What-is-new.pdf>. Acesso em: 28 de jun. de 2021.

LÜKE, W. G.; NASCIMENTO, A. F. **Gestão da Informação de Infraestrutura e edificações no Setor Público**. Autodesk University Brasil, 2012.

LUTZ, M. **Learning Python: Powerful Object-Oriented Programming**. 4ª ed. O'Reilly Media, 2009. ISBN: 978-0-596-15806-4.

LUTZ, M. **Programming Python: Powerful Object-Oriented Programming**. 4ª ed. O'Reilly Media, 2011. ISBN: 978-0596158101.

MASSÉ, M. **REST API: Design Rulebook**. 1ª ed. Sebastopol, California: O'Reilly Media, 2011. ISBN: 1449310508.

MATTOS, Aldo Dórea. **Planejamento e controle de obras**. 1.ed São Paulo: Pini, 2010.

MENDES, A. C. B. **Estudo aprofundado sobre o formato universal para troca de informações BIM IFC: estrutura do IFC para BIM 4D – planejamento e controle**. 2016. 117 f. Trabalho de Graduação (Graduação em Engenharia Civil) – Faculdade de Engenharia do Campus de Guaratinguetá, Universidade Estadual Paulista, Guaratinguetá, 2016.

MICROSOFT. **O que é o Windows Presentation Foundation (WPF)?**. 2021a. Online. Disponível em: <<https://docs.microsoft.com/pt-br/visualstudio/designers/getting-started-with-wpf?view=vs-2022>>. Acesso em: 06 de dez. de 2021.

MICROSOFT. **XAML in WPF**. 2021b. Online. Disponível em: <<https://docs.microsoft.com/en-us/dotnet/desktop/wpf/advanced/xaml-in-wpf?view=netframeworkdesktop-4.8>>. Acesso em: 06 de dez. de 2021.

MICROSOFT. **XAML overview (WPF .NET)**. 2021c. Online. Disponível em: <<https://docs.microsoft.com/en-us/dotnet/desktop/wpf/xaml/?view=netdesktop-6.0&viewFallbackFrom=netdesktop-5.0>>. Acesso em: 06 de dez. de 2021.

MÜLLER, A. C.; GUIDO, S. **Introduction to Machine Learning with Python: A Guide for Data Scientists**. 1ª ed. O'Reilly Media, 2016. ISBN: 978-1-449-36941-5.

PINHO, S. M. F. **O Modelo IFC como Agente de Interoperabilidade – Aplicação ao domínio das estruturas**. 2013. 155 f. Dissertação (Mestrado em Engenharia Civil). Faculdade de Engenharia, Universidade do Porto, Porto, 2013. Disponível em <<https://repositorio-aberto.up.pt/handle/10216/67927>>. Acesso em: 26 de jun. de 2021.

PUGA, S.; RISSETTI, G. **Lógica de programação e estruturas de dados com aplicações em Java**. 2ª ed. São Paulo: Pearson Prentice Hall, 2009. ISBN 978-85-7605-207-4.

PYREVIT. **pyRevit**. ©2021. Online. Disponível em: <<https://www.notion.so/pyrevitlabs/pyRevit-bd907d6292ed4ce997c46e84b6ef67a0>>. Acesso em: 04 de dez. de 2021.

RIBEIRO, D. C. **Avaliação da Aplicabilidade do IPD em Portugal**. 2012. 125 f. Dissertação (Mestrado em Engenharia Civil) – Departamento de Engenharia Civil, Faculdade de Engenharia da Universidade do Porto, Porto, Portugal, 2012. Disponível em: <https://paginas.fe.up.pt/~gequaltec/w/images/Tese_DavidRibeiro.pdf>. Acesso em: 26 de jul. de 2021

RIBEIRO, G. D. N. **Aplicação de ferramentas BIM em um projeto de cobertura do estádio Professor Dário Rodrigues Leite**. 2015. 70 f. Trabalho de Conclusão de Curso (Graduação em Engenharia Civil) – Faculdade de Engenharia de Guaratinguetá, Universidade Estadual Paulista, Guaratinguetá, São Paulo, 2015. Disponível em: <<https://repositorio.unesp.br/handle/11449/139232>>. Acesso em 26 de jul. de 2021.

RODRIGUES, L. F.; JESUS, R. A.; SCHÜTZER, K. **Indústria 4.0 – Uma revisão da Literatura**. 2016. Revista de Ciência & Tecnologia - V.19, p. 33–45, 2016. Disponível em: <<https://www.metodista.br/revistas/revistas-unimep/index.php/cienciatecnologia/article/view/3176>>. Acesso em 13 de jul. de 2021.

ROGGIA, L.; FUENTES, R. C. **Automação Industrial**. 2016. 102 f. Colégio Técnico Industrial de Santa Maria – Universidade Federal de Santa Maria, Rio Grande do Sul. Rede e-Tec Brasil, 2016. ISBN: 978-85-9450-001-4.

SCHWAB, K. **The Fourth Industrial Revolution**. 2016. 172 f. Cologny/Geneva, Switzerland: World Economic Forum. ISBN-10: 1944835016.

SENA, P. C. P. **Automação de processos de projeto e programação em BIM: Dynamo, Python e C#**. Dissertação (Mestrado em Arquitetura e Urbanismo) – Instituto de Arquitetura e Urbanismo, Universidade de São Paulo, São Carlos, 2019. Disponível em: <<https://www.teses.usp.br/teses/disponiveis/102/102131/tde-12032020-144132/pt-br.php>>. Acesso em: 22 de jun. de 2021.

SILVA, K. R. **Interoperabilidade entre Software de Projeto Estrutural com a Plataforma BIM**. 2019. 87 f. Monografia (Graduação em Engenharia Civil). Departamento de Engenharia Civil e Ambiental, Universidade de Brasília, Brasília - DF. Disponível em: <<https://bdm.unb.br/handle/10483/24301>>. Acesso em: 23 de jun. de 2021.

SILVA, Paula Heloisa da; CRIPPA, Julianna; SCHEER, Sergio. **BIM 4D no planejamento de obras: detalhamento, benefícios e dificuldades**. PARC Pesquisa em Arquitetura e Construção, Campinas, SP, v. 10, p. e019010, fev. 2019. ISSN 1980-6809. Disponível em: <<https://periodicos.sbu.unicamp.br/ojs/index.php/parc/article/view/8650258>>. Acesso em: 14 mai. 2021. doi: <https://doi.org/10.20396/parc.v10i0.8650258>.

TAMMIK, J. **RevitLookup: Interactive Revit `RFA` and `RVT` project database exploration tool to view and navigate BIM element parameters, properties and relationships**. Online. Disponível em: <<https://github.com/jeremytammik/RevitLookup>>. Acesso em: 04 de dez. de 2021.

THOMAS, D. **Revit Python Shell: An IronPython scripting environment for Autodesk Revit and Vasari**. Online. Disponível em: <<https://github.com/architecture-building-systems/revitpythonshell>>. Acesso em: 21 de jul. de 2021.

TOMAZ, P. **Cálculos Hidrológicos e Hidráulicos para Obras Municipais**. São Paulo: Editora Navegar, 2011.

VANDERPLAS, J. **Python Data Science Handbook: Essential Tools For Working With Data**. 1ª ed. O'Reilly Media, 2016. ISBN: 978-1491912058.