UNIVERSIDADE FEDERAL FLUMINENSE ESCOLA DE ENGENHARIA MESTRADO PROFISSIONAL EM MONTAGEM INDUSTRIAL

DANIEL LUIZ DE MATTOS NASCIMENTO

INTEROPERABILIDADE ENTRE SISTEMAS DE AUTOMAÇÃO DE PROJETOS EM TUBULAÇÕES INDUSTRIAIS UTILIZANDO INDUSTRY FOUNDATION CLASSES (IFC)

DANIEL LUIZ DE MATTOS NASCIMENTO

INTEROPERABILIDADE ENTRE SISTEMAS DE AUTOMAÇÃO DE PROJETOS EM TUBULAÇÕES INDUSTRIAIS UTILIZANDO INDUSTRY FOUNDATION CLASSES (IFC)

Dissertação apresentada ao Mestrado Profissional em Montagem Industrial da Universidade Federal Fluminense, como requisito parcial para obtenção do título de Mestre em Montagem Industrial. Área de Concentração: Montagem Industrial.

Orientador:
Prof. Roger Matsumoto Moreira, Ph.D.
Coorientador:
Prof. André Maués Brabo Pereira, Dr.techn.

Ficha catalográfica automática - SDC/BEE

N244i Nascimento, Daniel Luiz de Mattos

Interoperabilidade entre Sistemas de Automação de Projetos em Tubulações Industriais Utilizando Industry Foundation Classes (IFC) / Daniel Luiz de Mattos Nascimento; Roger Matsumoto Moreira, orientador; André Maués Brabo Pereira, coorientador. Niterói, 2017. 120 f.: il.

Dissertação (mestrado profissional)-Universidade Federal Fluminense, Niterói, 2017.

1. IFC. 2. Interoperabilidade. 3. Montagem Industrial. 4. Análise de Flexibilidade. 5. Produção intelectual. I. Título II. Moreira, Roger Matsumoto, orientador. III. Pereira, André Maués Brabo, coorientador. IV. Universidade Federal Fluminense. Escola de Engenharia.

CDD -

DANIEL LUIZ DE MATTOS NASCIMENTO

INTEROPERABILIDADE ENTRE SISTEMAS DE AUTOMAÇÃO DE PROJETOS EM TUBULAÇÕES INDUSTRIAIS UTILIZANDO INDUSTRY FOUNDATION CLASSES (IFC)

Dissertação apresentada ao Mestrado Profissional em Montagem Industrial da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Mestre. Área de Concentração Montagem Industrial.

Aprovada em 17 de abril de 2017.

BANCA EXAMINADORA
BLL.
Prof. Ph.D. Roger Matsumoto Moreira Presidente
/ Orientador
Universidade Federal Fluminense
Ad'M'Bi Pa
Prof. Dr. Tech. André Maués Brabo Pereira
Orientador
Universidade Federal Fluminense
Éliane M. b. louralto
Prof ^a . D.Sc. Eliane Maria Lopes Carvalho
Universidade Federal Fluminense
.151
Prof. Ph.D, Luiz Fernando Campos Ramos Martha
Pontificia Universidade Católica do Rio de Janeiro

AGRADECIMENTOS

À minha Família pela compreensão sobre o aspecto da dedicação necessária para alcançar este objetivo;

Aos meus amigos, sem eles não haveria aprendizado e perseverança para concluir os objetivos;

À Universidade Federal Fluminense e aos professores do Mestrado Profissional em Montagem Industrial:

Aos meus orientadores professor Roger Matsumoto Moreira e professor André Maués Brabo Pereira pelos ensinamentos e estímulo para alcançar os resultados esperados;

À minha gerência no Instituto Tecgraf/PUC-Rio ao nome de Eduardo Thadeu Leite Corseuil pela compreensão, orientação e apoio na dedicação ao mestrado;

À Pontifícia Universidade Católica do Rio de Janeiro e aos pesquisadores do Instituto Tecgraf pelo apoio e motivação para dedicar-se ao mestrado, especificamente agradecer a professora Geiza Maria Hamazaki da Silva e ao pesquisador Nicolas Alexandros Papadopoulos;

À Petrobras por permitir trabalhar em diversos projetos, os quais me capacitaram para realizar esta dissertação, propondo novas tecnologias para o setor de construção e montagem industrial.

À PUCRS por propiciar conhecimento em diversas disciplinas, as quais me capacitaram para realizar dissertação no setor de construção e montagem industrial;

À AVEVA pela disponibilização das licenças acadêmicas do PDMS na UFF, propiciando capacitar diversos alunos e fomentar a melhoria contínua do seu uso na indústria de óleo e gás.

LISTA DE SIGLAS E ABREVIATURAS

AECO Arquitetura, Engenharia, Construção e Operação

BIM Building Information Modeling

CAE Computer Aided Engineering

C&M Construção e Montagem

FEL Front End Loading

CAD

GUID Global Unique ID

IAI International Alliance for Interoperability

Computer Aided Design

IDM Information Delivery Manuals

IFC Industry Foundation Classes

IFD International Framework for Dictionaries

PMI Project Management Institute

PDMS Plant Design Management System

RESUMO

O presente trabalho tem por objetivo testar, implementar e analisar a interoperabilidade entre sistemas de engenharia, comumente empregados no setor de óleo e gás, utilizando o formato Industry Foundation Classes (IFC), amplamente difundido na indústria da construção civil, partindo da proposição de um modelo de maturidade para projetos de tubulações. A interoperabilidade sintática é verificada, com foco nos softwares comumente empregados no projeto executivo de tubulações industriais e estruturas metálicas. Inicialmente, propõe-se modelo de maturidade e fluxo de informações para projetos de tubulações, buscando otimizar recursos e aumentar a maturidade de projetos liberados para fabricação e montagem industrial. São implementadas duas soluções para o fluxo contínuo de informações, uma para o projeto de suportação e outra para a análise de flexibilidade de tubulações. Essas soluções realizam conversões das informações necessárias à análise estrutural dos suportes e de flexibilidade das tubulações, via formato IFC (Industry Foundation Classes) e XML (eXtensible Markup Language), respectivamente. Finalmente, são realizados estudos empíricos das soluções em projetos de instalações industriais. A primeira utiliza o IFC para troca bidirecional de informações necessárias à análise estrutural da suportação e compartilha elementos do catálogo entre ferramentas Open BIM e CAD/CAE 3D vinculados à uma norma específica, adotando-se neste caso a American Institute of Steel Construction (AISC). A segunda verifica atributos mínimos e disponibiliza informações para análise de flexibilidade no formato XML. Após testes das soluções desenvolvidas é possível perceber a viabilidade de mudança no fluxo de trabalho atual para projetar suportes estruturais com o IFC, no entanto, deve-se atentar para os tradutores das ferramentas existentes. Além disso, destaca-se a necessidade de mudança nos hábitos dos projetistas a fim de criar catálogos de componentes para modelagem 3D que atendam aos requisitos mínimos da análise de flexibilidade.

Palavras-chave: IFC; Interoperabilidade; Montagem Industrial; Análise de Flexibilidade; Análise Estrutural.

ABSTRACT

The present work aims to test, implement and analyze the interoperability between engineering systems, commonly employed in the oil and gas sector, using the Industry Foundation Classes (IFC) format, widely diffused in the construction industry, starting from the proposition of a maturity model for pipeline projects. Syntactic interoperability is verified, focusing on the software commonly used in the executive design of industrial pipes and metal structures. Initially, a maturity model and information flow for pipeline projects are proposed, seeking to optimize resources and increase the maturity of projects released for manufacturing and industrial assembly. Two solutions are implemented for the continuous flow of information, one for piping support design and the second for the analysis of piping flexibility. These solutions perform the necessary information conversions to enable the structural analysis of piping support and flexibility, via Industry Foundation Classes (IFC) and XML (eXtensible Markup Language) formats, respectively. Finally, empirical studies of the solutions in industrial plant projects are carried out. The first one uses IFC to bidirectionally exchange information needed to conduct piping support structural analysis and share catalog elements between Open BIM and CAD/CAE 3D tools linked to a standard, which in this case was the American Institute of Steel Construction (AISC). The second verifies minimum attributes and provides information for flexibility analysis in XML format. After testing the developed solutions, it is possible to perceive the feasibility of changing the current workflow to design structural supports with the IFC, however, one must pay attention to the translators of the existing tools. Furthermore, there is a need to change the habits of designers in order to create catalogues of 3D modeling components that meet the minimum requirements of flexibility analysis.

Keywords: IFC; Interoperability; Industrial Construction; Flexibility Analysis; Structural Analysis.

SUMÁRIO

1. IN	NTRODUÇÃO	10
1.1	MOTIVAÇÃO	12
1.2	OBJETIVO	14
1.2.1	1 OBJETIVO GERAL	14
1.2.2	2 OBJETIVOS ESPECÍFICOS	14
1.3	QUESTÕES DA PESQUISA	14
1.4 I	DELIMITAÇÃO DA PESQUISA	15
1.5 I	ESTRUTURA DA DISSERTAÇÃO	17
2. M	IETODOLOGIA DA PESQUISA	18
2.1.	DIAGNÓSTICO DE PROBLEMAS EM PROJETOS DE TUBULAÇÕES INDUSTRIAIS	18
2.2.	MÉTODOS DA PESQUISA	22
	EVISÃO DA LITERATURA	
3.1.	PROJETO E ANÁLISES EM TUBULACOES INDUSTRIAIS	25
	O CONCEITO DE BIM (<i>BUILDING INFORMATION MODELING</i>) E A SINERGIA COM	26
3.3.	INTEROPERABILIDADE EM SISTEMAS DE ENGENHARIA	31
3.4.	INDUSTRY FOUNDATION CLASSES (IFC)	35
3.5.	TRABALHOS RELACIONADOS SOBRE O IFC	43
3.6.	FERRAMENTAS E LINGUAGENS EXPLORADAS	44
	ROPOSIÇÃO DE FLUXOS DE TRABALHO PARA AUMENTAR A MATURIDADE EM PROJETOS TUBULAÇÕES INDUSTRIAIS	
4.1 I	MODELO DE MATURIDADE	47
4.2 I	PROCESSO COLABORATIVO	50
4.3 I	FLUXOS DE INFORMAÇÕES	51
5. E	STUDO EXPERIMENTAL	55
5.1	TESTES DE INTEROPERABILIDADE ENTRE FERRAMENTAS OPEN BIM	56
5.2.	DESENVOLVIMENTO DE CONVERSOR BIDIRECIONAL PARA PROJETO DE SUPORTES	56
5.3.	DESENVOLVIMENTO DE UMA SOLUÇÃO PARA ANÁLISE DE FLEXIBILIDADE	76
5.4.	ESTUDO EMPÍRICOS	82

5.4.1. ESTUDO EMPIRICO I - APLICAÇAO DA SOLUÇAO COMPUTACIONAL EM PROJETO DE SUPORTAÇÃO	82
5.4.2. ESTUDO EMPÍRICO II – APLICAÇÃO DA SOLUÇÃO COMPUTACIONAL EM PROJETO DE SUPORTAÇÃO	
5.4.3 ESTUDO EMPÍRICO III - APLICAÇÃO DA SOLUÇÃO COMPUTACIONAL PARA ANÁLISE I FLEXIBILIDADE	
6. CONCLUSÃO E PESQUISAS FUTURAS	94
REFERÊNCIAS	97
Apêndice I – Código de Programação na linguagem Python para Projeto de Suportes	105
Apêndice II – Código de Programação na linguagem PML para análise de flexibilidade	114

LISTA DE FIGURAS

Figura 1: Disciplinas interligadas pela modelagem da informação de uma planta industrial	11
Figura 2: Delimitação do escopo da pesquisa.	16
Figura 3: Gráfico de Pareto com a frequência de problemas que compartilham da mesma causa nos proj tubulações.	
Figura 4: Métodos de pesquisa qualitativos e quantitativos.	23
Figura 5: Etapas para triangulação da pesquisa.	24
Figura 6: Estágios de maturidade na implementação de BIM	30
Figura 7: Integração ponto-a-ponto (P2P) entre sistemas de engenharia	33
Figura 8: Interoperabilidade com formato neutro e quantidade de transformações	34
Figura 9: Especificação da entidade IfcProduct na Linguagem EXPRESS	38
Figura 10: Parte da especificação da entidade IfcObjectPlacement na Linguagem EXPRESS	38
Figura 11: Especificação da entidade IfcProductRepresentation na Linguagem EXPRESS	39
Figura 12: Viga modelada e exportada em formato IFC para ilustração.	39
Figura 13: Trecho de código utilizado na análise da instância do objeto definido na linha #29 IFCLOCALPLACEMENT	41
Figura 14: Diagrama de maturidade de projeto 3D para tubulações industriais	48
Figura 15: Fluxo de Informações para projeto 3D de tubulações industriais.	52
Figura 16: Conjunto de suportes modelados no Autodesk Revit.	56
Figura 17: Conjunto de suportes importados no Autodesk Robot via IFC 2x3.	57
Figura 18: Conjunto de suportes importados no Robot via integração direta entre os softwares	58
Figura 19: Conjunto de suportes importados via IFC 2x3 no SAP 2000	60
Figura 20: Conjunto de suportes no BIM Vision.	61
Figura 21: Conjunto de suportes no Solibri Model Viewer.	61
Figura 22: Suporte L64X6 criado no Revit, com material definido "Steel ASTM A36"	62
Figura 23: Arquivo IFC 2x3 com o suporte L64X6 criado no Revit.	62
Figura 24: Suporte L64X6 criado no Autodesk Revit, sem informações de materiais	63
Figura 25: Escrita de um formato Json a partir do IFC 2x3 utilizado em testes.	63
Figura 26: Conversão para o formato Json a partir do IFC 2x3 utilizado em testes.	64
Figura 27: Integração ponto-a-ponto à esquerda e interoperabilidade sintática com IFC à direita	65

Figura 28: Detalhamento das informações mais relevantes de uma viga no formato Datal	66
Figura 29: Arquitetura do conversor bidirecional entre Datal e IFC.	68
Figura 30: Viga modelada para ilustração sobre o processo de leitura de uma viga no IFC.	70
Figura 31: Interface para conversão de IFC para Datal.	74
Figura 32: Interface para conversão de Datal para IFC.	75
Figura 33: Interface para importação da norma utilizada na conversão.	75
Figura 34: Arquitetura da implementação e proposição de formato IFC para análise de flexibilidade	76
Figura 35: Hierarquia da extração de dados do PDMS via PML.	78
Figura 36: Diagrama de hierarquia do processo de extração e verificação de informações	78
Figura 37: Taxonomia de informações para análise de flexibilidade de tubulações industriais	80
Figura 38: Projeto de tubulação com modelagem básica (etapa 1), inclusão de acessórios (etapa 2) e proje suportação (etapa 3) realizados	
Figura 39: Processamento da conversão do Datal para o IFC 2x3	83
Figura 40: Resultado da conversão do Datal para o IFC 2x3	83
Figura 41: Modelo analítico da suportação estrutural para unidade de retortagem no Scia Engineering	84
Figura 42: Visualização 3D do suporte estrutural no BIM Vision	84
Figura 43: Visualização 3D do suporte estrutural com projeções no espaço no BIM Vision	85
Figura 44: Processamento do suporte estrutural do Scia Engineering via IFC para o Datal	85
Figura 45: Importação do Datal com suporte estrutural aprovado no PDMS	86
Figura 46: Visão geral dos suportes estruturais para tubulações e eletrocalhas no PDMS	87
Figura 47: Processamento de arquivo Datal para IFC 2x3, contendo um suporte estrutural	87
Figura 48: Suporte estrutural de tubulações e eletrocalhas no visualizador BIM Vision	88
Figura 49: Resultado da importação do arquivo IFC gerado, no Solibri Model Viewer	88
Figura 50: Resultado da importação e deformação do suporte no SAP 2000	89
Figura 51: Processo de retorno ao PDMS a partir da conversão do IFC 2x3 em Datal	90
Figura 52: Retorno dos dados do suporte de tubulações e eletrocalhas para o PDMS	90
Figura 53: Linha de tubulação selecionada para extração de dados para análise de flexibilidade	91
Figura 54: XML extraído de um projeto real em uma unidade de destilação à vácuo	92
Figura 55: Diagrama do algoritmo de conversão bidirecional em datal ou IFC	105
Figura 56: Diagrama do algoritmo de exportação do PDMS para o XML	114

LISTA DE TABELAS

Tabela 1: Principais problemas ou pontos de atenção identificados em projeto de tubulaçõesErre	o! Indicador
não definido.	
Tabela 2: Atributos instanciados na classe <i>Ifcbeam</i> .	40
Tabela 3: Relatório do Autodesk Robot com análise dos suportes importados via IFC	59
Tabela 4: Classes, funções e implementação da leitura do formato IFC pelo conversor	72
Tabela 5: Tabela de atributos para conversão bidirecional do Datal ou IFC.	73
Tabela 6: Análise de importação e exportação de suportes modelados no <i>Autodesk Revit</i>	94

1. INTRODUÇÃO

O ciclo de vida na indústria da construção envolve diversas tecnologias, processos, materiais e pessoas, sendo importante a convergência das partes envolvidas pelo interesse global do empreendimento, que pode ser obtida através da transferência de conhecimento com sequenciamento adequado para garantir um fluxo contínuo e confiável de informações.

Entretanto, a heterogeneidade de disciplinas, tecnologias e padronizações torna a transferência do conhecimento e a interligação dos fluxos de trabalho desafios proeminentes para o sucesso de empreendimentos. Atualmente em diversos tipos de contratos, as disciplinas de projeto ainda são mensuradas físico-financeiramente por emissão de documentos, o que dificulta o fluxo da informação, gerando inconsistências entre projetos e, consequentemente, retrabalhos na fase de construção. A convergência de trabalho das diversas disciplinas com interesses mútuos e/ou individuais é um ponto de atenção para cumprir metas e minimizar desperdícios em projetos de engenharia.

Uma solução para este problema é a implantação da interoperabilidade entre os diversos sistemas, por meio da adoção de uma linguagem padrão e o mapeamento das trocas de dados necessárias, com objetivo de interligar harmonicamente fluxos de trabalho. Vale a pena salientar que a falta de interoperabilidade entre sistemas foi destacada como causa fundamental da ineficiência de empreendimentos nos Estados Unidos da América (EUA) em 2002, contabilizando perdas estimadas em 15,8 bilhões de dólares anualmente (Gallaher *et al.*, 2004).

Por compartilhar problemas semelhantes com a automação de projetos em instalações industriais, a construção civil inovou ao desenvolver e implantar o conceito de *Building Information Modeling* (BIM), que envolve metodologias e tecnologias utilizadas na produção, comunicação, análise e simulação de dados, com destaque para modelos 3D.

A primeira referência de BIM é realizada no trabalho de Nederveen e Tolman (1992), a qual pode ser vislumbrada como uma nova geração das ferramentas CAD/CAE, contendo solução envolvente frente às diversas tecnologias, disciplinas e fases do ciclo de vida de um empreendimento na indústria da construção civil. Analogamente aos preceitos de BIM, Oliveira (1995) conceitua a automação de projetos como tecnologia CAD/CAE para gestão da informação que integra todos os envolvidos em um empreendimento, buscando melhorar a comunicação, a documentação e a produtividade em prol da otimização dos processos de engenharia.

A Figura 1 ilustra as diversas disciplinas envolvidas no projeto de uma instalação industrial, demonstrando que o BIM pode ser empregado para alcançar um fluxo de trabalho simultâneo e colaborativo. No projeto 3D, pode-se analisar que esta metodologia de trabalho permite a integração em um ambiente único das disciplinas de tubulações, civil e estruturas, elétrica, instrumentação, equipamentos e suportação.

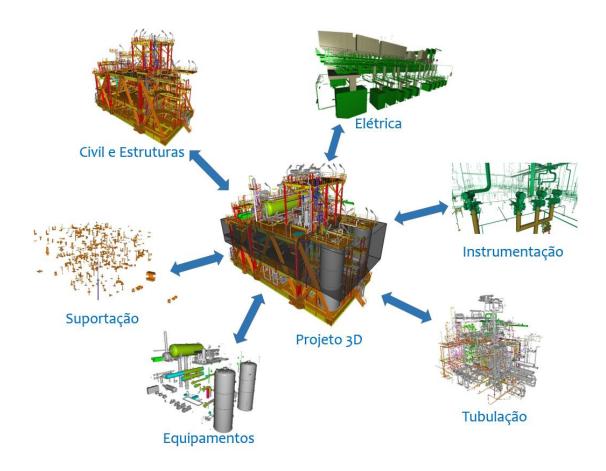


Figura 1: Disciplinas interligadas pela modelagem da informação de uma planta industrial.

Para prover a comunicação entre as diversas disciplinas, de forma a permitir a análise e simulação de modelos 3D, é necessário a existência de uma linguagem padrão. Neste contexto, o *Industry Foundation Classes* (IFC), que é um formato aberto, neutro e com especificações padronizadas, de acordo com a metodologia BIM, torna-se um relevante candidato para seu uso na construção e montagem industrial. Este padrão foi inicialmente criado visando propiciar a interoperabilidade de sistemas da indústria de Arquitetura, Engenharia, Construção e Operação (AECO). Contudo, nos dias de hoje, diversos sistemas que contemplam funcionalidades interdisciplinares em projetos de engenharia estão certificados no padrão IFC.

1.1 MOTIVAÇÃO

A recorrente falta de informações integradas e consistentes entre disciplinas para o planejamento e controle da montagem industrial proporciona diversos tipos de retrabalho na execução, perdas de material, diminuição da produtividade e imprecisão entre o previsto e realizado em um empreendimento. Sobretudo, os efeitos percebidos impactam no aumento do prazo e custo, e na diminuição da qualidade e segurança nos projetos deste setor.

Analogamente, no setor da construção civil são compartilhados diversos problemas de natureza semelhante, afetando a análise e interpretação das informações nos empreendimentos. Segundo Governo de Santa Catarina (2015), por exemplo, destacam-se os seguintes problemas: a falta de transparência, redundância e duplicação de informações, sistemas sem compatibilidade com padrões não proprietários, arquivos nomeados sem padronização da estrutura de dados, modificação de coordenadas e escalas, entre outros.

Na metodologia tradicional dos projetos executivos, no qual utiliza-se modelagem 2D, as informações presentes nos documentos não são interligadas entre si, desconectando fluxos de trabalho principalmente entre projeto e construção (Papadopoulos *et al.*, 2017). Segundo Eastman *et al.* (2011), o processo de construção e entrega de um empreendimento permanece descentralizado e baseado em documentações impressas no papel como forma de comunicação entre os envolvidos. Atrasos, retrabalhos e eventuais ações judiciais estão no cotidiano das equipes, devido aos documentos gerados com erros e omissões nas informações de projeto.

Como já foi colocado na seção anterior, esses problemas podem ser creditados, fundamentalmente, pela inexistência da interoperabilidade para interligar fluxos de trabalho padronizados na metodologia tradicional. Segundo Gallaher *et al.* (2004) esse fato gera inúmeras perdas em projetos de engenharia, onde se destacam:

- O custo de redigitar manualmente dados;
- O custo de verificar informações;
- Perdas para os clientes e os consumidores devido ao atraso na disponibilidade de produtos e serviços;
- Recursos ociosos pelo atraso do início da operação;
- Perda de lucro devido ao atraso da receita;

- Os custos de atraso do início da operação;
- O custo do retrabalho entre projeto e construção;
- O custo de aquisição, manutenção e treinamentos em sistemas de engenharia redundantes.

No Brasil, a adoção do conceito BIM faz parte do esforço de redução destas perdas, principalmente na iniciativa privada, embora algumas medidas governamentais estão sendo tomadas para apoiá-la. Em 2010, a Associação Brasileira de Normas Técnicas (ABNT) nomeou uma comissão para estudar e desenvolver técnicas para a geração de normas direcionadas à implementação de BIM. Trata-se da ABNT/CEE-134 - Comissão de Estudo Especial de Modelagem de Informação da Construção. Esta comissão foi estabelecida em 2010 e elaborou desde então cinco normas¹ que já foram colocadas em vigor:

- ABNT NBR 15965-7:2015 Sistema de classificação da informação da construção,
 Parte 7: Informação da construção;
- ABNT NBR 15965-3:2014 Sistema de classificação da informação da construção,
 Parte 3: Processos da construção;
- ABNT NBR 15965-2:2012 Sistema de classificação da informação da construção,
 Parte 2: Características dos objetos da construção;
- ABNT NBR 15965-1:2011 Sistema de classificação da informação da construção,
 Parte 1: Terminologia e estrutura;
- ABNT NBR ISO 15965-3: 2009 Construção de edificação Organização de informação da construção, Parte 2: Estrutura para classificação de informação.

Mediante o exposto, a dificuldade da obtenção de informações interligadas e confiáveis entre equipes no ciclo de vida de um empreendimento na construção civil é análoga à área industrial. Ambos segmentos são impactados pelo uso inadequado da interoperabilidade, sendo que a construção civil adquiriu maior maturidade sintática no curto prazo com a criação e ampla adoção do padrão IFC por fornecedores de softwares e clientes. Neste contexto, o objetivo deste estudo é apresentado no tópico seguinte.

_

¹ http://www.abntcatalogo.com.br/normagrid.aspx acessado em 10/01/2017.

1.2 OBJETIVO

1.2.1 OBJETIVO GERAL

O presente trabalho tem por objetivo geral testar, implementar e analisar a interoperabilidade entre sistemas de engenharia, comumente empregados no setor de óleo e gás, utilizando o formato *Industry Foundation Classes* (IFC), amplamente difundido na indústria da construção civil, partindo da proposição de um modelo de maturidade para projetos de tubulações.

1.2.2 OBJETIVOS ESPECÍFICOS

A proposta do presente trabalho refere-se à utilização do formato IFC no setor de óleo e gás, envolvendo a área de projetos de tubulações industriais. Busca contribuir para obtenção dos seguintes benefícios: maior consistência dos projetos de tubulações industriais; integração entre o planejamento da montagem de tubulações com a instalação dos suportes; redução da quantidade de revisões dos documentos de projeto; e garantia da disponibilidade dos suportes no momento de necessidade da montagem de tubulação. Os objetivos específicos do trabalho são os seguintes:

- Propor um fluxo de trabalho para tubulações industriais, com objetivo de aumentar a maturidade e assertividade do projeto executivo;
- Realizar testes sobre a efetividade da interoperabilidade utilizando o IFC entre sistemas da AECO;
- Implementar soluções computacionais e estudos empíricos para avaliar na prática a interoperação de informações em projetos de suportação e extração de dados para análise de flexibilidade;
- Verificar a efetividade dos conversores bidirecionais desenvolvidos para o IFC e XML, respectivamente; e
- Propor uma taxonomia (organização da informação com relações hierárquicas entre os objetos) de forma adequada para permitir a interoperabilidade entre modelos 3D e ferramentas que realizam análise de flexibilidade em tubulações industriais, possibilitando um padrão sintático para propor extensões no IFC.

1.3 QUESTÕES DA PESQUISA

Verifica-se que muitas metodologias e funcionalidades desenvolvidas na engenharia civil podem ter aplicabilidades em outras engenharias. Sendo assim, por meio da aplicação dos conceitos de BIM, este trabalho tem o objetivo de responder a seguinte questão:

Como a área de tubulações industriais pode ser beneficiada com o padrão de interoperabilidade fornecido pelo *Industry Foundation Classes*?

Para responder esta pergunta, pretende-se avaliar a hipótese de que estas novas metodologias e funcionalidades possam ser aplicadas e gerar benefícios no setor de óleo e gás. Desta forma, descrevem-se as seguintes perguntas secundárias:

- Como pode ser utilizado o formato IFC no segmento de óleo e gás para realizar análise estrutural da suportação anexada à tubulação industrial, interligando as disciplinas de estruturas e tubulações?
- Quais informações devem ser utilizadas por meio de uma taxonomia para inclusão de classes dedicadas para tubulações industriais e análise de flexibilidade no formato IFC?
- Como melhorar o processo de maturidade do projeto 3D de tubulações industriais, utilizando a interoperabilidade sintática entre sistemas de engenharia?
- Quais benefícios, barreiras e limitações que foram encontrados, para orientar trabalhos futuros de natureza semelhante?

1.4 DELIMITAÇÃO DA PESQUISA

O estudo concentrou-se em utilizar a interoperabilidade por meio do IFC diretamente em projetos de suportação e extração com conversão para o formato XML na análise de flexibilidade em tubulações industriais. A ferramenta CAD/CAE 3D utilizada como fonte e/ou modelo central para aplicação das soluções desenvolvidas em projetos de plantas industriais é o *Plant Design Management System* (PDMS).

Propõe-se um fluxo de trabalho no projeto 3D para projetar tubulações industriais, realiza-se implementação de conversor bidirecional da ferramenta de projeto 3D (PDMS) para o formato IFC e formato XML. No estudo experimental é delimitado um escopo de teste em linhas de tubulação com seus respectivos suportes, provenientes de projetos reais de plantas industriais modelados em PDMS.

Esta pesquisa tem caráter documental, teórico e experimental, e descreve os pontos fortes e fracos da interligação entre fluxos de trabalho em projetos para construção civil e tubulações industriais. A partir disso, por meio da interoperabilidade é possível promover o gerenciamento visual do projeto executivo, de forma colaborativa, envolvendo todas as equipes necessárias em projetos de tubulações industriais. Na Figura 2, pode-se verificar o delineamento da pesquisa na conversão bidirecional de informações entre o projeto 3D e ferramentas de análise para calcular a tubulação como elemento estrutural e realizar análises de flexibilidade.

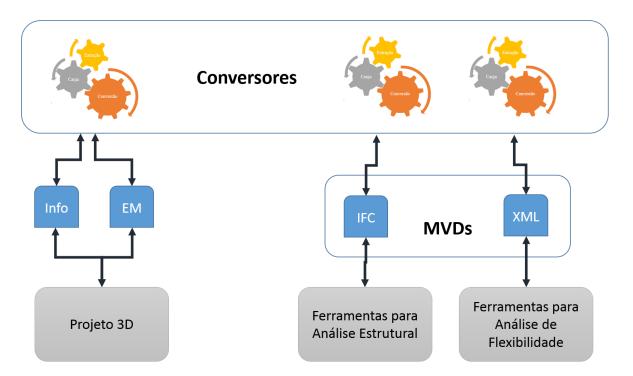


Figura 2: Delimitação do escopo da pesquisa.

Nesta pesquisa é estabelecido um fluxo de trabalho para projetar tubulações industriais, utilizando conceitos da interoperabilidade sintática, por meio de extração e tradução de informações do projeto de suportação estrutural e de flexibilidade das tubulações industriais para os formatos IFC e XML, respectivamente.

1.5 ESTRUTURA DA DISSERTAÇÃO

Capítulo 1 - Inicia-se este capítulo com considerações iniciais da pesquisa, contendo contexto e histórico, motivação contendo problemas e tendências, objetivos geral e especifico, as questões da pesquisa, delimitação da pesquisa e estrutura da dissertação;

Capítulo 2 - Metodologia da pesquisa contendo diagnóstico do problema e os métodos para solução dos problemas identificados;

Capítulo 3 - Revisão da literatura sobre tubulações industriais, interoperabilidade, BIM e IFC. Neste capítulo as palavras-chave citadas são exploradas com objetivo de conceituar e verificar trabalhos relacionados, a partir de uma revisão sistemática da literatura;

Capítulo 4 – Proposição de modelo de maturidade para projetos de tubulações industriais, utilizando o modelo 3D como elemento central, descrição de processo colaborativo para elaboração do projeto e utilização da interoperabilidade sintática neste fluxo de trabalho, com destaque para o IFC;

Capítulo 5 – Estudo experimental contendo testes entre diversos sistemas com formato de arquivo IFC, implementação computacional e aplicação das soluções desenvolvidas em projetos reais;

Capítulo 6 – Conclusão por meio de triangulação teórica e experimental para responder as perguntas ou hipóteses formuladas, também propor sugestões para pesquisas futuras sobre o ambiente explorado nesta presente dissertação.

2. METODOLOGIA DA PESQUISA

Nesse capítulo, apresenta-se um diagnóstico dos problemas ou pontos de atenção percebidos em projetos de engenharia, identificando suas causas e efeitos para nortear a presente dissertação. Em seguida são estabelecidos os métodos de pesquisa, explorando diferentes procedimentos em busca de soluções das causas mais frequentes, dentro do escopo delimitado em projetos de tubulações industriais.

2.1. DIAGNÓSTICO DE PROBLEMAS EM PROJETOS DE TUBULAÇÕES INDUSTRIAIS

Os problemas norteadores da pesquisa são obtidos a partir de um levantamento de dados no setor de óleo e gás. Na análise em projetos de tubulação para o refino de petróleo, realizaramse entrevistas seriadas por meio de um grupo focal para coleta, tratamento e apresentação dos principais problemas, identificação das causas e dos efeitos percebidos (Neves, 1996; Ribeiro, 2003; Tortorella *et al.*, 2008).

Os eventos foram realizados entre outubro de 2013 e março de 2014, selecionou-se a partir de um universo com 42 especialistas, uma amostra intencional de 12 colaboradores mais representativos no tema, os quais eram responsáveis pelo projeto de uma instalação industrial. Essa fração amostral de coordenadores, consultores e engenheiros representam 28,57% do total de especialistas na fase de projeto executivo deste empreendimento. Apresenta-se abaixo o critério de seleção da amostra do grupo focal:

- Objetivo: obter informações sobre problemas, identificar suas causas fundamentais e analisar lições aprendidas dos efeitos percebidos em projetos de tubulações;
- Unidades de referência: colaboradores que trabalhavam no projeto, construção e montagem de um complexo petroquímico para o refino de petróleo;
- Unidade informante: coordenadores, consultores e engenheiros;
- Unidade de análise: projetos de tubulação para construção e montagem de instalações industriais do complexo petroquímico;
- Unidade amostral: pessoas que continham maior conhecimento no tema central da investigação e ocupavam cargos de liderança no empreendimento.

Nesse contexto a partir da aplicação de um questionário semiestruturado, foram identificados 29 problemas relativos à fase de projeto executivo. Esses problemas tornaram-se norteadores para presente dissertação, uma vez que expõe a necessidade de melhorias na indústria. A Tabela 1 apresenta as informações dos principais problemas categorizados por palavras-chave e descrição (problema ou ponto de atenção).

Tabela 1: Principais problemas ou pontos de atenção identificados em projeto de tubulações.

ID	Palavra-chave	Problema ou ponto de atenção
1	Fornecedores	Falta de informações dos fornecedores de equipamentos.
2	Recursos humanos	Pouca experiência das projetistas contratadas.
3	Análise de Flexibilidade	Atraso na elaboração da memória do cálculo de flexibilidade.
4	Automação de	Pouca utilização das ferramentas de automação
7	Projetos Projetos	de projetos.
5	110]0:05	O planejamento e a execução do projeto
	Construtibilidade	executivo devem ser realizados seguindo o que foi estabelecido nos estudos de construtibilidade, pois o projeto deve priorizar a sequência de montagem (por exemplo, do norte a sul ou de leste para oeste).
6	Construtibilidade	A equipe multidisciplinar de construção e montagem, que participou dos estudos de construtibilidade, deve acompanhar o desenvolvimento do projeto executivo orientando sobre boas práticas construtivas e níveis de detalhes.
7	Construtibilidade	Em muitos casos, a equipe de projeto vem poucas vezes em campo tirar dúvidas, verificar interferências e interfaces.
8	Recursos humanos	Equipe insuficiente para a execução do projeto.
9	Recursos humanos	Profissionais da contratante com pouca experiência.
10	Critério de medição	Baixo percentual na Estrutura Analítica de Projeto (EAP) para modelos 3D, minimizando a importância das empresas na elaboração, controle e consistência do projeto 3D.
11	Automação de Projetos	Falta de designação de profissional específico para fiscalização do modelo 3D e verificação das interfaces com demais grupos e disciplinas de projeto executivo.
12	Automação de Projetos	Falta de adequação dos dados de projeto para possibilitar interligação entre ferramenta de controle da fabricação e montagem de tubulações e modelo 3D com níveis de confiabilidade aceitáveis.

Tabela 1 (cont.).

13	Planejamento e	Projeto e montagem de tubulações sem
	Controle	integração com a instalação dos suportes.
14	Revisão de	Quantidade elevada de revisões de documentos
	Documentos	com materiais em fabricação.
15	Automação de	Falta de consistência entre linha de tubulação,
	Projetos	suportes e modelo 3D.
16	Construtibilidade	Falta definição de juntas e trechos da linha de tubulação para fabricação no isométrico.
17	Construtibilidade	Falta definição de juntas na emissão do isométrico.
18	Automação de Projetos	A emissão de um novo isométrico para construção/fabricação deve ser extraída a partir do Modelo 3D.
19		Demora na resposta da solicitação de informação
	Matriz de	técnica ou respostas não conclusivas do projeto
	atribuições	para construção e montagem, tornando-as
		ineficientes.
20	Aprovado para	Fluxogramas em revisão, gerando retrabalho no
	Construção	projeto executivo de tubulação.
21	Planejamento e	D 111 ~ '~ 1 11 1 1
	Controle	Demora na liberação e revisões da lista de linhas.
22	M-4- 1-1	Atualização do modelo 3D não acompanha o
	Metodologia	avanço do projeto realizado em documentos 2D,
23	Automosão do	gerando retrabalho.
23	Automação de	Isométricos sem representação das quebras de
24	Projetos	especificações técnicas de tubulação.
24	Metodologia	Não definição da classe de inspeção no projeto executivo.
25	Metodologia	Falta da indicação da condição de pintura e
		isolamento térmico no projeto executivo.
26	Metodologia	Falta de padronização dos códigos de materiais de tubulação no projeto básico.
27	Matadalasia	Referências topográficas do projeto
1	Metodologia	inconsistentes com a real.
28	Automação de	Interferências no projeto 3D entre tubulações,
	Projetos	suportes e equipamentos.
29	Integração	Indefinições de interfaces entre tubulações nos limites de bateria.

Os problemas ou pontos de atenção levantados (citados na Tabela 1) foram apresentados em um evento para discussão dos resultados, com foco na identificação de suas causas e seus efeitos na construção e montagem do complexo petroquímico. Na discussão, realizou-se a alocação de cada um dos 29 pontos levantados a uma causa fundamental. Os participantes discutiram sobre cada aspecto, estimulado pelo moderador do evento, até alcançar o consenso.

A metodologia de grupo focal tem como objetivo coletar a percepção das pessoas acerca do tema central de uma investigação, preferencialmente quando se existe homogeneidade entre os envolvidos e representatividade da amostra (Oliveira e Freitas, 1998). Nesse contexto, ao final dos eventos, segundo a percepção dos envolvidos foram detectadas 13 causas fundamentais. A quantidade de problemas por categorias que compartilham da mesma causa é apresentada na Figura 3.

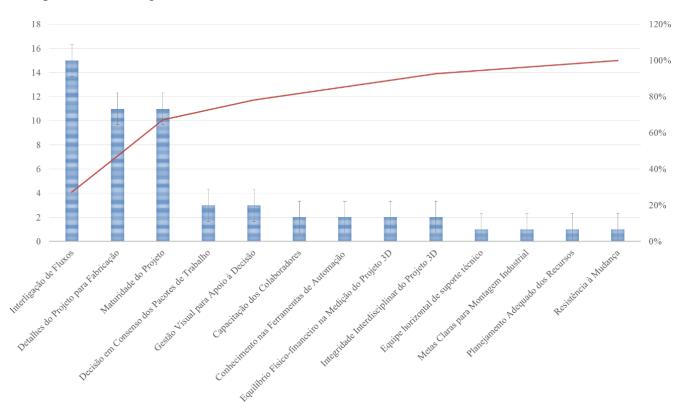


Figura 3: Gráfico de Pareto com a frequência de problemas que compartilham da mesma causa nos projetos de tubulações.

Pode-se observar que as ineficiências na interligação de fluxos de trabalho, no detalhamento do projeto para a fabricação, na maturidade de projeto, na decisão em consenso dos pacotes de trabalho, na gestão visual e na integridade interdisciplinar do projeto 3D compreendem 82% das causas fundamentais na fase de projeto executivo. A partir das causas dos problemas ou pontos atenção em projetos de tubulações identificados, percebeu-se os principais efeitos na construção e montagem. Esses efeitos foram destacados como motivação para estabelecer os objetivos específicos, citados no primeiro capítulo. Na seção seguinte são apresentados os métodos de pesquisa utilizados para solução das causas mais frequentes identificadas e diminuição dos efeitos negativos gerados na construção e montagem.

2.2. MÉTODOS DA PESQUISA

O método de pesquisa escolhido é exploratório, descritivo e explicativo. Sobretudo busca diminuir as inconsistências na interligação de informações entre sistemas e propõe um processo de trabalho que busca contribuir para interdisciplinaridade. O objetivo do método é facilitar a troca de dados e aumentar a maturidade de cada linha de tubulação projetada com foco na diminuição do retrabalho.

Na primeira etapa é realizada uma pesquisa documental e bibliográfica de tubulações industriais, em seguida verifica-se a sinergia entre automação de projeto e BIM, a interoperabilidade de sistemas e o conceito do IFC. Nesta pesquisa a análise dos dados foi realizada de forma intuitiva e indutiva, verificando cada conceito nas bases de pesquisas nacionais e internacionais com seus resultados apresentados no capítulo 3.

Na segunda etapa, para as causas dos problemas percebidos (explicitados na seção anterior), propõe-se uma solução metodológica de tratamento tanto para o aumento da maturidade quanto para interligação do fluxo de informações, aplicável em projetos de tubulação. Os resultados obtidos podem ser encontrados no capítulo 4.

Na terceira etapa, desenvolvem-se duas soluções tecnológicas que exploram a interoperabilidade. Utilizam-se as soluções desenvolvidas no âmbito da automação de projetos tanto em projeto de suportes quanto na análise de flexibilidade. Os resultados desta etapa são apresentados no capítulo 5.

Na quarta etapa discute-se sobre experimentos acerca das soluções tecnológicas desenvolvidas e aplicadas, literatura, documentos e sua efetividade de aplicação no modelo proposto para otimizar a engenharia colaborativa. O resultado é encontrado no capítulo 6.

Portanto, mediante o exposto nas etapas citadas desta dissertação, explora-se a abordagem qualitativa com pesquisa documental e bibliográfica, também há rodadas de grupos focais para definir a situação problema e análise das causas fundamentais. Na etapa quantitativa, realiza-se por meio de experimentos tecnológicos o desenvolvimento de duas soluções para automação de projetos de tubulações industriais, utilizando a interoperabilidade em implantação de funcionalidades BIM no setor de óleo e gás: (I) fluxo *just-in-time* de informações do projeto 3D e (II) *Jidoka* com verificações semiautomáticas para o projeto de suportes estruturais e consistência de dados para análises de flexibilidade, definindo o estudo como quali-quantitativo, conforme é demonstrado na Figura 4.

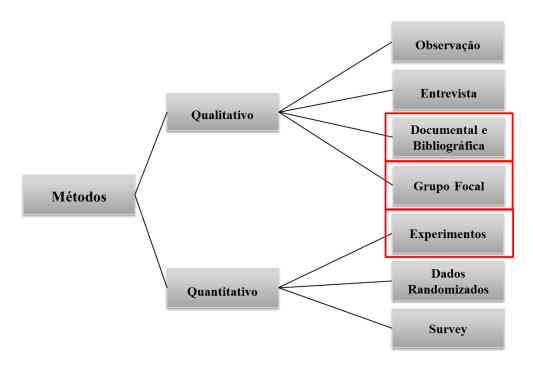


Figura 4: Métodos de pesquisa qualitativos e quantitativos.

Fonte: adaptado de Briner e Denyer (2012).

A abordagem também é exploratória porque objetiva levantar informações mais relevantes sobre interoperabilidade, *Industry Foundation Classes* (IFC), funcionalidades BIM e tubulações industriais, compatibilizando e, quando necessário, propondo extensões no IFC para projeto de tubulações no segmento de montagem industrial. É descritiva porque busca revelar como as informações podem ser apresentadas com seus reflexos pertinentes à construção e montagem industrial.

Os dados qualitativos foram coletados para diagnóstico dos problemas ou pontos de atenção em projetos de tubulações na implementação de empreendimentos industriais. Os dados quantitativos são extraídos de projetos reais de plantas industriais através de soluções por softwares desenvolvidas em escala piloto nesta pesquisa, buscando interligar ferramentas de automação de projetos dos empreendimentos, por meio de uma linguagem padrão no nível de interoperabilidade sintática. Utilizam-se o IFC e o XML, respectivamente, para o projeto de suportes estruturais e análise de flexibilidade. Para tanto, a metodologia da pesquisa é composta pelas seguintes etapas ilustradas na Figura 5.

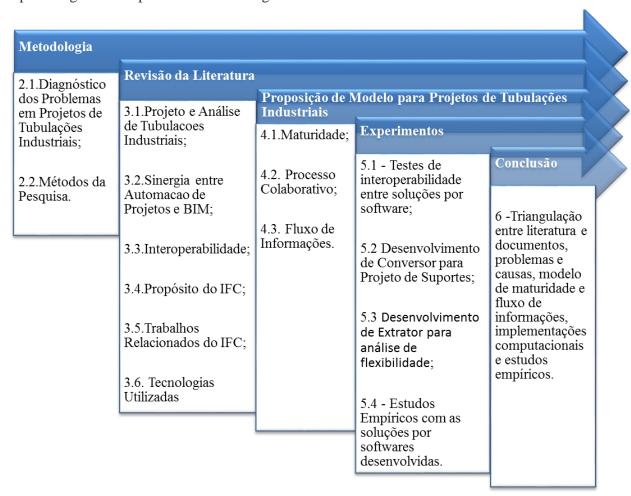


Figura 5: Etapas para triangulação da pesquisa.

O estudo é considerado válido na medida em que os dados são obtidos de diferentes procedimentos (literatura, documentos, reuniões e experimentos), o que constitui uma triangulação, evitando a subjetividade do pesquisador e garantindo a qualidade dos resultados (Greenhalgh e Taylor, 1997; Yin, 2014). No capítulo seguinte é apresentada a revisão da literatura.

3. REVISÃO DA LITERATURA

A revisão da literatura tem o objetivo principal de investigar documentos técnicos e artigos científicos para fornecer um levantamento abrangente acerca do tema pesquisado, interligando os pontos de vistas de diferentes autores. Desta forma, foram verificados trabalhos da engenharia civil e informática relacionados com a automação de projetos, sendo possível fazer a correlação entre eles com foco em projetos de tubulações para plantas industriais. Neste capítulo, definem-se conceitos que serão explorados nos experimentos, analisando como os pesquisadores registram suas contribuições e quais áreas do conhecimento necessitam de maior ênfase.

3.1. PROJETO E ANÁLISES EM TUBULAÇÕES INDUSTRIAIS

No ciclo de vida de plantas industriais, as tubulações têm maior peso físico-financeiro, quando comparadas as demais disciplinas de um empreendimento. Credita-se o alto impacto das tubulações à interligação entre equipamentos e o uso de aço em diversas especificações, atendendo as necessidades determinadas pelo processo petroquímico.

Segundo Telles (2004) a tubulação pode ser considerada como um elemento estrutural. Cada segmento de uma linha de tubulação é submetido a diversas cargas atuantes, que geralmente agem sobre suportes e/ou pontos de ancoragem. As linhas podem estar ligadas ou não a equipamentos. Caso positivo, os esforços são limitados pela tensão máxima admissível. Ademais, deve-se entender a relação entre as cargas atuantes no sistema e a resistência dos materiais. Com esta finalidade, deve-se analisar cada trecho de apoio realizando cálculo estrutural, definindo os suportes com dimensões e materiais mais adequados ao projeto.

Ainda de acordo com Telles (2004), as cargas que atuam sobre uma tubulação são: pressão interna proveniente do fluido, pressão externa, peso próprio e sobrecargas diversas. As sobrecargas agem sobre o peso de estruturas apoiadas no tubo e suas relações causais podem ser categorizadas como: dilatações térmicas, movimento de pontos extremos da tubulação causadas por dilatação de outras tubulações ou dilatação de equipamentos ligados à tubulação, atrito da tubulação nos suportes, ações dinâmicas provenientes de movimento do fluido da tubulação, ações dinâmicas externas, vibrações, reações das juntas de expansão, tensões decorrentes da montagem e desnivelamento dos suportes ou equipamentos.

A tubulação, como um conjunto de elementos estruturais, contém tensões primárias e secundárias. As tensões primárias formam um sistema envolvendo diversas cargas atuantes sobre a tubulação e suportes, sendo necessário a realização de análises estruturais, visando atingir o equilíbrio estático para conformidade legal. As tensões secundárias são provenientes de efeitos secundários que atuam sobre elementos estruturais, tais como, dilatações, movimentações e contrações. A análise das tensões secundárias deve ser realizada no cálculo de flexibilidade, que é um ponto crítico de atenção para qualidade e efetividade do projeto de tubulações.

Na seção seguinte são apresentados trabalhos sobre a sinergia entre a automação de projetos e o *Building Information Modeling* (BIM), com objetivo de estabelecer conceitos sobre o assunto e analisar pontos de contribuição científica buscando aumentar a qualidade e, por conseguinte, maturidade em projetos de tubulações industriais.

3.2. O CONCEITO DE BIM (*BUILDING INFORMATION MODELING*) E A SINERGIA COM AUTOMAÇÃO DE PROJETOS

O conceito de BIM preconiza funcionalidades que visam interligar fluxos de trabalho dentro de uma representação visual conforme o projeto será ou foi construído para análises, comunicação, simulações e verificações em prol do gerenciamento efetivo de um empreendimento na indústria da Arquitetura, Engenharia, Construção e Operação (AECO). Essas funcionalidades propiciam automação dos processos de engenharia, maior confiabilidade das informações e democratização do projeto 3D, buscando eliminar tarefas manuais e repetitivas ao longo do ciclo de vida de um ativo (Sacks *et al.*, 2010; Eastman *et al.*, 2011; Park *et al.*, 2013).

Segundo Silva (2013) algumas etapas dos processos de engenharia em plantas industriais se mostram ineficientes devido à realização de tarefas repetitivas relacionadas à grande quantidade de áreas e/ou disciplinas, sistemas e informações. Nesse contexto, a automação de projetos busca eliminar atividades manuais e repetitivas que diminuem o tempo disponível para resolução de problemas técnicos. Sendo assim, cada vez mais tecnologias são consumidas em plantas industriais para desenvolver metodologias e otimizar os processos de engenharia (Rodrigues, 2008; Machado e Abreu, 2011; Silva, 2013).

De acordo com Babič *et al.* (2010), as sinergias entre funcionalidades de BIM provenientes da construção civil e o setor de montagem industrial foram encontradas por meio de customizações realizadas nas ferramentas da automação de projetos. Ao utilizar essas customizações, os autores destacam benefícios na integração entre o projeto 3D e a construção. Contudo, afirmam que obtiveram insucesso na adoção destas funcionalidades pelos parceiros e fornecedores, tornando a utilização limitada ao ambiente interno e ocasionando diversos tipos de retrabalhos. A causa principal pode ser creditada à falta de interoperabilidade nas trocas de informações entre fluxos de trabalho. BIM é conceituado de forma abrangente e estruturada, compartilhando funcionalidades correlatas à automação de projetos. De acordo com Sacks *et al.* (2010) e Eastman *et al.* (2011), BIM é definido pelas seguintes funcionalidades:

- Visualização 3D (por estética e avaliação funcional);
- Geração rápida de múltiplas alternativas de projeto;
- Uso de dados do modelo para análise preditiva (análise preditiva do desempenho, estimativa de custos automatizada e avaliação de conformidade ao valor do cliente);
- Manutenção de informação e integridade do modelo (fonte de informação única, verificações automatizadas de regras das disciplinas envolvidas);
- Geração automática de desenhos e documentos;
- Colaboração entre projeto e construção (edição multiusuário de um modelo de disciplina única, visualização multiusuário de modelos multidisciplinares separados ou mesclados);
- Geração e avaliação rápidas de alternativas para planos de construção (geração automática de tarefas para construção, simulação do processo de construção e visualização multidimensional de dados;
- Comunicação baseada em objeto online/eletrônico (visualização do processo, comunicação online de informações de produto e processo); e
- Transferência de informação direta para apoio à fabricação controlada por computador.

O setor de construção civil tem aplicado soluções para automação de projetos com a estruturação e progressiva adoção das funcionalidades BIM por governos e iniciativa privada (Andrade e Ruschel, 2009). Nesse contexto, Sacks *et al.* (2010) concluem que as funcionalidades de BIM efetuam mudanças fundamentais na Arquitetura, Engenharia, Construção e Operação (AECO). Desta forma, os autores afirmam que um novo paradigma é creditado ao BIM, pois proporciona uma transformadora modelagem e interligação de informações propiciando o gerenciamento visual do empreendimento. A partir desse ambiente, pode-se realizar simulações e análises apoiando à decisão na definição do planejamento e controle.

Entretanto, diversos pesquisadores identificam barreiras nos processos de engenharia dos empreendimentos na indústria da construção para implantação de BIM (Sacks *et al.*, 2010; Linderoth, 2010; Zhiliang *et al.*, 2011; Eadie *et al.*, 2013). Entre elas, destacam-se limitação de softwares, falta de interoperabilidade e capacitação nas tecnologias disponíveis (Gu e London, 2010; Singh *et al.*, 2011; Redmond *et al.*, 2012; Grilo e Jardim-Gonçalves, 2010; Wang *et al.*, 2013; Park *et al.*, 2013; Kim *et al.*, 2013; Xiong *et al.*, 2013; Davies e Harty, 2013). Por fim, as competências das pessoas, que envolvem atitudes, habilidades, conhecimento e resistência à mudança, são fatores críticos de sucesso levantados por alguns autores para implantação de BIM (Isikdag e Underwood, 2010; Shen *et al.*, 2012; Zhang *et al.*, 2013).

Nesse contexto, Arayici *et al.* (2011) relatam que os sete pilares de uma estratégia de implantação BIM são: eliminar o desperdício; aumentar o retorno de informações sobre o processo; alcançar o consenso; entrega rápida; construir com integridade; capacitar a equipe; e ver o todo. Os autores afirmam que a implementação de BIM deve ter uma abordagem do nível estratégico para o operacional, a fim de: i) envolver oficialmente as pessoas na adoção, ii) assegurar o aumento das competências, da compreensão das pessoas sobre o assunto e que empresas construam suas capacidades, iii) aplicar estratégias para gerenciamento de mudanças bem-sucedidas, e iv) diminuir qualquer potencial resistência à mudança.

Allison (2010) afirma que alguns dos potenciais benefícios do uso de BIM para a gestão da construção são percebidos nos seguintes aspectos:

- Organização do cronograma físico-financeiro;
- Trabalho colaborativo com a equipe do projeto;
- Melhoria na contratação e controle dos contratados e/ou subcontratados;

- Melhoria nos pedidos de informações e ordens de alteração;
- Melhor registro de conhecimento tácito e satisfação do cliente;
- Aumento na margem do projeto;
- Facilidade de troca de produto e informações entre contratante e contratada; e
- Crescimento da efetividade no gerenciamento de projetos.

Mediante aos benefícios comprovados, o Reino Unido está efetuando as mudanças necessárias para a adoção de novas tecnologias que superem pontos críticos na indústria da construção, através de uma iniciativa governamental. Estas mudanças ocorreram para permitir a implementação de BIM por meio de um plano estratégico, oficializado e publicado pelo gabinete em 31 de maio de 2011, concretizando a obrigatoriedade, a partir de 2016, do processo BIM colaborativo em 3D com informações de projeto e cadeia de suprimento para todos os empreendimentos (com todas as informações sobre projetos e ativos, documentação e dados eletrônicos).

De acordo com documento do governo britânico denominado *HM Government Construction 2025* (2013), o país está investindo em capacidade de produção, definindo ações necessárias para que o governo e a indústria possam seguir no mesmo sentido da criação de oportunidades para o setor de construção, tornando-se um líder mundial em BIM. No documento é estabelecido um modelo de como a indústria e o governo devem trabalhar juntos para colocar o Reino Unido na vanguarda da construção mundial.

Khosrowshahi e Arayici (2012) afirmam que a implementação de BIM é uma tarefa árdua de gestão da mudança em processos, tecnologias e pessoas, envolvendo diversas áreas do conhecimento na indústria da construção. Segundo os autores existem três estágios de maturidade na implementação de BIM, apresentados na Figura 6.

Estágio 1 – Modelagem baseada em objetos. Nesta fase a documentação é composta por um modelo 3D orientado a objetos, contendo geração automática e coordenação dos pacotes de informações com dados básicos coletados a partir do modelo 3D, tais como, planos 2D, elevações, seções e quantitativos em geral. Os relatórios podem ser pós-processados por meio de ágeis visualizações 3D. Neste estágio a colaboração é assíncrona e estabelecida por meio de arquivos.

Estágio 2- Modelagem baseada em colaboração. Neste estágio a informação deve ser compartilhada e trocável, o projeto deve conter duas dimensões adicionais para realizar a visualização 4D (3D mais a dimensão de tempo) e visualização 5D (4D mais a dimensão de custo), gerar relatório de entregas a partir do escopo analisado, realizar análise de interferências físicas entre disciplinas e utilizar a colaboração assíncrona.

Estágio 3 – Práticas integradas, provendo a visualização multidimensional de dados para realizar análises complexas em estágios iniciais do empreendimento, como por exemplo, de sustentabilidade, construtibilidade, ciclo de vida, custo, entre outras. As disciplinas multidisciplinares devem utilizar o mesmo modelo 3D, explorando a interoperabilidade e modelo federado (modelo central contendo todas disciplinas atualizadas), fluxos de trabalho enxutos e simplificados, comunicação sincronizada e ambientes com vários servidores para colaboração entre equipes.

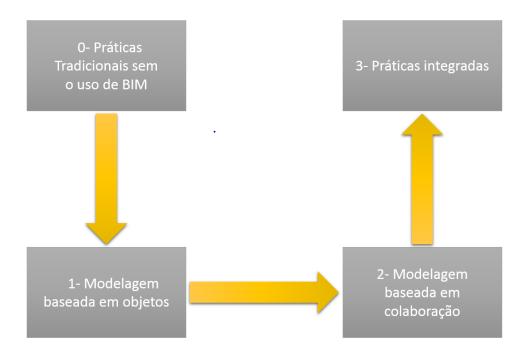


Figura 6: Estágios de maturidade na implementação de BIM

Fonte: Adaptado de Khosrowshahi e Arayici (2012).

Os estágios da maturidade de BIM podem ser utilizados como ferramenta de *benchmarking* para comparação de regiões e medição da maturidade das suas implementações (Khosrowshahi e Arayici, 2012).

A estratégia apresentada no documento *HM Government Construction 2025* (2013) tem o objetivo de atingir a redução de 33% do custo inicial da construção e de todo custo no ciclo de vida dos ativos; a redução de 50% de tempo total da concepção até o encerramento de uma nova construção e restauração dos ativos; a redução de 50% dos gases de efeito estufa no ambiente construído; e a redução de 50% na balança comercial entre o total de exportação e importação de produtos e materiais de construção a partir de 2016 até 2025.

A contextualização das pesquisas de BIM e automação de projetos permitiu verificar correlação entre seus objetivos. Entretanto, diferentemente da automação de projetos o BIM tem sido implantado na indústria AECO como uma estratégia que envolve iniciativa público e privada, organizando uma metodologia de evolução da capacidade de sua utilização, em um ambiente composto por universidades e empresas em prol da melhoria contínua e incremental. Uma barreira fundamental identificada em ambas iniciativas é a falta de interoperabilidade entre os diversos sistemas e processos, gerando perdas em todo ciclo de vida de um ativo. Na seção seguinte é conceituada e contextualizada pesquisas acerca da interoperabilidade entre sistemas de engenharia na construção civil e montagem industrial.

3.3. INTEROPERABILIDADE EM SISTEMAS DE ENGENHARIA

Eastman *et al.* (2011) conceitua a interoperabilidade como a habilidade de trocar informações entre aplicações, interligando fluxos de trabalho harmonicamente e algumas vezes propiciando automação de processos. A interoperabilidade permite que sistemas se comuniquem efetivamente, mesmo sendo de diferentes tipos e/ou produzidos por diversos fornecedores. Segundo Pereira *et al.* (2011), a interoperabilidade é uma solução que reforça a necessidade de interligações entre sistemas de engenharia, evitando as perdas que se tem com o uso desconexo das ferramentas de automação de projetos em fluxos de trabalho de um empreendimento.

A interoperabilidade, segundo Tolk e Muguira (2003), pode ser classificada em seis níveis de maturidade crescentes:

 Nível 1 (interoperabilidade técnica): promove relação técnica e permite trocar dados entre sistemas (por exemplo: protocolos, bits e bytes, simbologia de informações, conectividade de rede e outros);

- Nível 2 (interoperabilidade sintática): linguagem e formato padronizados para troca de informações entre sistemas com dados estruturados;
- Nível 3 (interoperabilidade semântica): sistemas interoperacionais trocam um conjunto de termos. As conformidades dos objetos interoperáveis podem ser analisadas semanticamente com foco na engenharia do conhecimento;
- Nível 4 (interoperabilidade pragmática): sistemas são cientes do contexto das informações, tais como, estados, processos e significados dos objetos envolvidos na troca entre fluxos de trabalho;
- Nível 5 (interoperabilidade dinâmica): sistemas interoperacionais capazes de reorientar produção e consumo das informações de acordo com mudanças percebidas dentro dos fluxos de trabalho, gerando efeitos nos dados e na troca de informações;
- Nível 6 (interoperabilidade conceitual): interoperando neste nível é alcançada a troca total de informações, processos, contextos, premissas de modelagem e, portanto, conhecimento.

Existe uma diferença que deve ser destacada entre integração e interoperabilidade. A integração, segundo Wang *et al.* (2009), trata-se da conexão de informações entre dois ou mais sistemas sem a necessidade de adotar um formato de arquivo padronizado. Entretanto, Tolk e Muguira (2003) afirmam que a interoperabilidade se baseia na adoção de uma linguagem e formato de arquivo padronizados, estruturados para troca de informações necessárias entre as soluções por software que incorporam o padrão escolhido.

A partir disso, *links* bidirecionais das aplicações para o formato padronizado são requisitos importantes para a implantação da interoperabilidade. De acordo com diversos autores, conversores são capazes de traduzir de uma linguagem para outra e podem ser utilizados para promover compatibilidade entre dois ou mais padrões de arquivo (Grilo e Jardim-Goncalves, 2010; Hu e Zhang, 2011; Zhiliang *et al.*, 2011; Eastman *et al.*, 2011; Venugopal *et al.*, 2015).

Na Figura 7, é demonstrado em um esquemático o conceito da integração ponto a ponto (P2P). Trata-se da integração entre sistemas, sem a utilização da interoperabilidade. Percebe-se um número elevado de transformações à medida que aumenta a quantidade de sistemas interoperados para interligar fluxos de trabalho.

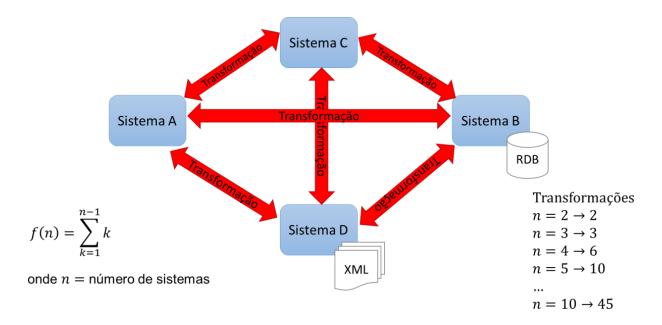


Figura 7: Integração ponto-a-ponto (P2P) entre sistemas de engenharia.

Fonte: Pereira e Clasen (2013).

A interoperabilidade sintática de sistemas preconiza minimamente a adoção de uma linguagem padrão estruturada, conforme apresentado na Figura 8. Esta solução diminui o número de transformações ponto a ponto, tornando as interações proporcionais ao número de sistemas interoperados. Existe a demanda por tradutores de todos os sistemas envolvidos para a linguagem padrão como requisito mínimo para atingir o nível 2 de maturidade em interoperabilidade.

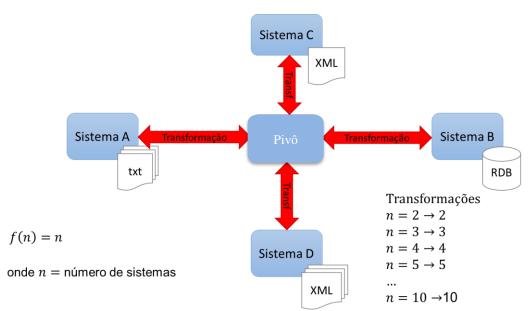


Figura 8: Interoperabilidade com formato neutro e quantidade de transformações.

Fonte: Adaptado de Pereira e Clasen (2013).

A interoperabilidade sintática propicia atendimento às necessidades de trocas de informações entre processos, tecnologias e pessoas, permitindo a contribuição de diferentes competências e tecnologias de toda a equipe envolvida em uma visão estratégica. De acordo com Santos (2008), os benefícios obtidos pela interoperabilidade entre fluxos de trabalho e sistemas são:

- Aumento da competitividade econômica;
- Redução do tempo para lançamentos de novas tecnologias no mercado;
- Redução de vulnerabilidade da infraestrutura;
- Expansão de mercados para as empresas;
- Diminuição dos custos de comunicação da cadeia de suprimentos;
- Fornecimento de acesso global para fornecedores de software.

De acordo com Gökçe *et al.* (2013), a interoperabilidade permite que governo e setor privado de um país possam trocar informações e produtos de forma estruturada, livre de fornecedores monopolistas no mercado, permitindo aumentar a governança em projetos. O novo processo de trabalho creditado à interoperabilidade, se implementado, não permite

direcionamento em contratos. Este processo aumentará a concorrência e inovação dos fornecedores, gerando maior competição no mercado entre empresas em prol da melhoria contínua para maximizar receitas e aumentar fidelização de seus clientes com melhores técnicas e qualidade.

Neste trabalho, é proposta a utilização do *Industry Foundation Classes* (IFC), como formato para padronização das informações em projetos de tubulação industrial na área de óleo e gás, propiciando aumento na qualidade e redução do retrabalho. A revisão da literatura do IFC será discutida na seção a seguir.

3.4. INDUSTRY FOUNDATION CLASSES (IFC)

De acordo com Eastman *et al.* (2010), IFC é um formato neutro e aberto proveniente do formato STEP (*Standard for the Exchange of Product Model Data*), que é reproduzível para contemplar os dados de um produto, e sua linguagem para descrição de metadados é a EXPRESS. O STEP surgiu em 1994 para interoperação entre modelos CAD, sendo motivado pela ineficiência das trocas de informações entre modelos baseados em primitivas geométricas. Nesse contexto, o IFC, criado pela *Building Smart International* (BSI) em 1997, propõe um esquema para modelagem conceitual dos dados relativos à geometria, georreferenciamento, plantas em 2D e engenharia (Liebich, 2008).

Ferreira et al. (2007) afirmam que o modelo dos dados segue o paradigma de orientação a objetos e é descrito utilizando a linguagem EXPRESS, na qual uma classe descreve uma entidade com seus atributos e relacionamentos, permitindo assim a instanciação de um objeto (por exemplo, uma viga). Este modelo é flexível, podendo ser expandido em futuras versões do formato IFC. A adoção do formato IFC para a troca e compartilhamento de informações entre as funcionalidades de BIM, em todo ciclo de vida, é vinculada à norma internacional ISO 16739 (2013). As informações interoperadas podem ser visualizadas, modificadas, analisadas e restritas a pacotes de informações contendo somente o que é necessário para interligar fluxos de trabalho entre equipes ou todo o projeto 3D. Além disso, o IFC pode ser visto como um formato padronizado para o livre acesso e transporte de informações entre soluções por software. Cabe destacar que a padronização é pertinente às informações interoperadas na indústria AECO e não à estrutura interna dos softwares.

Segundo Filho e Sheer (2009), a arquitetura do IFC é subdividida respectivamente da camada mais alta para a mais baixa: domínio (domain) - trata-se das disciplinas ou equipes de um empreendimento; interoperabilidade (interoperability) - permite estabelecer relações para troca de dados entre domínios; central (core) - refere-se às unidades comuns a todos domínios e relações para troca de dados; e recursos (resource) - descreve conceitos básicos a serem utilizados pelos níveis mais altos. Andrade e Ruschel (2009) definem uma classe como a descrição de um conjunto de objetos que compartilham atributos, operações, relacionamentos e semântica. Um domínio tem um conjunto de classes, como por exemplo, o domínio dos elementos estruturais que contém uma classe viga (IfcBeam) e pode-se relacionar a uma classe de material (IfcMaterial) que está na camada de recurso material.

Ademais, Filho e Sheer (2009) destacam que as entidades podem representar objetos, atributos de objetos e/ou relações entre objetos - *IfcObject, IfcPropertyDefinition, IfcRelationship*, respectivamente. As entidades no modelo do IFC segundo Liebich e Wix (2008) são detalhadas em sete tipos principais:

- Produto (*IfcProduct*): trata-se de objetos que podem ser alocados no projeto 3D em determinada coordenada global. São itens físicos e/ou produtos necessários a um projeto, podendo ser reutilizados ou elaborados internamente no catálogo;
- Processo (*IfcProcess*): refere-se a diagramas sequenciais e temporais que relacionam (*IfcRelAssignsToProcess*) tecnologias, pessoas e materiais em prol de ações (projetar, suprir, construir, montar, manter produtos, entre outros) convergentes ao objetivo estratégico. A relação estabelecida entre predecessoras e sucessoras (*IfcRelSequence*) permite atribuir recursos necessários para cumprir cada etapa dentro de um processo;
- Controle (*IfcControl*): pode-se controlar e/ou aplicar restrições nos objetos para cumprir requisitos dentro de um processo. Esta entidade é classificada como uma especificação, regulamento, cronograma de custos ou outro requisito aplicado a um produto e/ou processo que uma vez atribuído deve ser cumprido. Os controles são destinados a produtos, processos e objetos utilizando o relacionamento entre eles na camada de interoperabilidade (*IfcRelAssignsToControl*);

- Recursos (*IfcResource*): utiliza-se para representar insumos necessários a uma atividade do projeto, podendo ser relacionada a um produto (*IfcProduct*) ou ator envolvido (*IfcActor*) através do relacionamento (*IfcRelAssignsToResource*) na camada de interoperabilidade;
- Ator (*IfcActor*): define os atores ou colaboradores envolvidos em um projeto durante seu ciclo de vida, permitindo associá-los a uma pessoa e organização para facilitar alocação dos recursos em objetos no projeto 3D;
- Projetos (*IfcProject*): adotam-se padrões relativos a um projeto, tais como, unidades
 de medida, sistema de coordenadas globais, dimensão do espaço de coordenadas,
 precisão usada dentro das representações geométricas, e indicação do norte real em
 relação ao sistema de coordenadas global;
- Grupos (*IfcGroup*): trata-se de uma coleção lógica de objetos, classificados ou alocados em categorias, contendo aspectos não geométricos para organizar as informações.

Na definição do modelo conceitual dos dados, as entidades são definidas por meio de um contexto, contendo relacionamento hierárquico (subtipo ou supertipo), que interligam seus atributos ou relacionam-se entre entidades. Para ilustrar a navegação e interpretação das entidades, será analisada a entidade *IfcProduct*², a qual é subtipo da entidade *IfcObject*³, herdando assim todas suas características descritas pelos seus atributos e relacionamentos.

A especificação da entidade *IfcProduct*, ver figura 9, além das características herdadas da entidade *IfcObject*, contém dois atributos opcionais, um para representação das coordenadas do objeto, *ObjectPlacement* da classe *IfcObjectPlacement*, e outro com a representação relativa ao material do produto, *Representation* da classe *IfcProductRepresentation*.

http://www.buildingsmart-tech.org/ifc/IFC2x4/rc2/html/schema/ifckernel/lexical/ifcobject.htm acessado em 10/01/2017

_

² http://www.buildingsmart-tech.org/ifc/IFC2x4/rc2/html/schema/ifckernel/lexical/ifcproduct.htm acessado em 10/01/2017

ENTITY IfcProduct

ABSTRACT SUPERTYPE OF (ONEOF (If cAnnotation, If cElement, If cGrid, If cPort, If cProxy, If cSpatial Element, IfcStructuralActivity, IfcStructuralItem))

SUBTYPE OF IfcObject;

ObjectPlacement: OPTIONAL IfcObjectPlacement; Representation: OPTIONAL IfcProductRepresentation;

END ENTITY;

Figura 9: Especificação da entidade *IfcProduct* na Linguagem EXPRESS.

Navegando pela definição dos atributos expostos na figura 9, de forma a compreender quais informações para definir um produto, será analisado a entidade *IfcObjectPlacement*⁴, a qual o atributo ObjectPlacement pertence. Na especificação desta entidade, tem-se que o ObjectPlacement deve ser um IfcGridPlacement ou IfcLocalPlacement, que são especializações da entidade IfcObjectPlacement. Tanto a classe IfcGridPlacement quanto a classe *IfcLocalPlacement* definem, de forma diferenciada, o sistema de coordenadas do objeto, conforme ilustra a Figura 10.

ENTITY IfcObjectPlacement

ABSTRACT SUPERTYPE OF(ONEOF(IfcGridPlacement, IfcLocalPlacement));

INVERSE

PlacesObject: SET[1:?] OF IfcProduct FOR ObjectPlacement;

ReferencedByPlacements: SET OF IfcLocalPlacement FOR PlacementRelTo;

END ENTITY;

Figura 10: Parte da especificação⁵ da entidade *IfcObjectPlacement* na Linguagem EXPRESS.

Continuando a análise da entidade IfcProduct, Figura 9, o atributo Representation da classe *IfcProductRepresentation*, Figura 11. pode uma instância de IfcProductDefinitionShape⁶ ou de IfcMaterialDefinitionRepresentation⁷, dependendo se a

4 http://www.buildingsmart-

tech.org/ifc/IFC2x4/alpha/html/ifcgeometricconstraintresource/lexical/ifcobjectplacement.htm acessado em

⁵ Por questões didáticas foi suprimido parte da especificação.

⁶ http://www.buildingsmarttech.org/ifc/IFC2x3/TC1/html/ifcrepresentationresource/lexical/ifcproductdefinitionshape.htm acessado em 10/01/2017

⁷ http://www.buildingsmart $tech. org/ifc/IFC2x\overline{3}/TC1/html/ifcrepresentation resource/lexical/ifc material definition representation. html/resource/lexical/ifc material definition representation and the second resource of the seco$ acessado em 10/01/2017

representação fizer referência a uma forma ou a múltiplas representações de um mesmo material, respectivamente.

```
ENTITY IfcProductRepresentation

SUPERTYPE OF(ONEOF( IfcProductDefinitionShape, IfcMaterialDefinitionRepresentation));

Name: OPTIONAL IfcLabel;

Description: OPTIONAL IfcText;

Representations: LIST[1: ?] OF IfcRepresentation;

END_ENTITY;
```

Figura 11: Especificação da entidade *IfcProductRepresentation* na Linguagem EXPRESS.

Vale salientar que o processo para análise de estrutura da entidade e de seus componentes deve ser realizado de forma recursiva, até encontrar a unidade básica da definição, neste caso o tipo ou estrutura de dados. Dado o conhecimento de como deve ser realizada a navegação na hierarquia das entidades/classes do padrão IFC, de forma a entender sua estrutura, em seguida é apresentado um trecho para instanciação das classes na definição de uma viga utilizando o IFC (Figura 12) e como deve ser realizado a leitura deste arquivo.

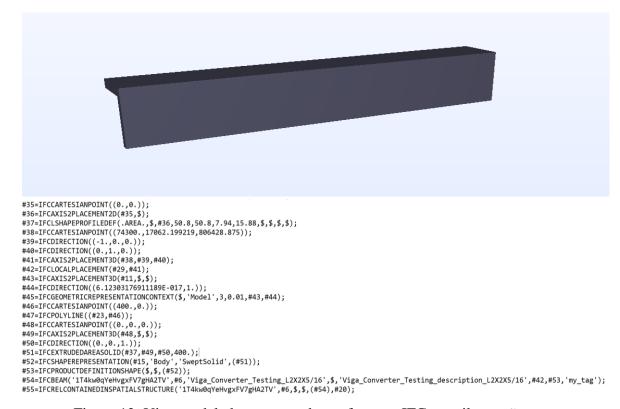


Figura 12: Viga modelada e exportada em formato IFC para ilustração.

A leitura deste arquivo, exposto na Figura 12 para a compreensão da instanciação de uma viga, deve ser iniciada na entidade *IfcBeam*, presente na linha #54, como descrito a seguir:

#54=IFCBEAM('3_3jFmqruHvhCiV7gHA2TV',#6,'Viga_Converter_Testing_L102X102X7.9',\$,'Viga_Converter_Testing_description_L102X102X7.9',#42,#53,'my_tag');

Esta entidade é composta de vários atributos, obrigatórios e opcionais. Entretanto será explicado apenas os utilizados neste exemplo. Os atributos são instanciados na seguinte ordem, ilustrados na Tabela 2.

Tabela 2: Atributos instanciados na classe Ifcbeam.

Atributos	Requisitos	Valor no exemplo	Descrição		
GUID	Obrigatório	'3_3jFmqruHvhCiV	Chave primária única de		
		7gHA2TV'	cada objeto no IFC.		
OwnerHistory	Obrigatório		Define toda história e		
		#6	identificação das		
			informações relacionadas ao		
			objeto.		
Name	Opcional	'Viga_Converter_Te	Define nome para um		
		sting_L102X102X7.	determinado tipo ou perfil		
		9'	de suporte.		
Description	Opcional	'Viga_Converter_Te	Descreve o detalhamento de		
		sting_description_L	informações acerca da		
		102X102X7.9'	descrição das características		
			de um perfil.		
ObjectType	Opcional	\$ (não preenchida)	Tipo de Objeto		
ObjectPlacement	Obrigatório		O posicionamento da viga		
		#42	relativo ao posicionamento		
			no projeto 3D do qual a viga		
			pertence.		
Representation	Obrigatório		Define todas as informações		
		#53	relevantes da forma na		
			classe <i>IfcProduct</i> . Permite		
			várias representações de		
			forma geométrica do mesmo		
_			produto.		
Tag	Opcional	'my_tag'	Tagueamento ou Identidade		
			de cada elemento		

Nota1: É interessante observar que a ligação entre as entidades é realizada através da referência à linha em que se encontra à instância da classe definida na especificação da entidade (por exemplo: #6 refere-se ao atributo *OwnerHistory*).

Continuando a análise do código presente na Figura 12, considera-se que o *IfcBeam* faz referência diretamente a instanciação de um *ObjectPlacement* que é especializado pela entidade *IfcLocalPlacement*, como permitido pela especificação Figura 10, presente na linha:

```
#42=IFCLOCALPLACEMENT (#29, #41).
```

Os atributos desta entidade definem a coordenada relativa ao objeto, linha #29 IFCLOCALPLACEMENT(\$,#26), e o sistema de coordenadas a ser aplicado na transformação a partir dos dados fornecidos/ no primeiro atributo, linha #41 IFCAXIS2PLACEMENT3D(#38,#39,#40). Considerando a aplicação sucessiva de referências às instâncias, tem-se que as definições presentes nas linhas: #23, #38, #39 e #40 possuem os valores numéricos sobre os quais serão realizados os cálculos para a representação e análise das características do objeto definido pelas entidades (Figura 13).

```
#29= IFCLOCALPLACEMENT($, #26);

#26 = IFCAXIS2PLACEMENT3D(#23,$,$);

#23 = IFCCARTESIANPOINT((0., 0., 1.));

#41= IFCAXIS2PLACEMENT3D(#38,# 39, #40);

#38 = IFCCARTESIANPOINT((310540., 298501.599998, 104880.));

#39= IFCDIRECTION((-1., 0., 0.));

#40= IFCDIRECTION((0., 1., 0.));
```

Figura 13: Trecho de código utilizado na análise da instância do objeto definido na linha #29 IFCLOCALPLACEMENT.

A leitura dos arquivos no padrão IFC deve ser realizado sobre as instâncias de forma recursiva até chegar ao elemento raiz, que possui dados geométricos ou não geométricos relativos às necessidades de engenharia (que pode ser, por exemplo, uma cadeia de caracteres ou um valor numérico).

O processo aplicado na leitura dos arquivos no padrão IFC para a compreensão da descrição dos objetos é o mesmo a ser aplicado na escrita utilizando este padrão, no qual torna-se essencial o conhecimento da especificação das entidades, das camadas superiores para as inferiores na hierarquia, para a possível instanciação de todas as entidades que compõem os objetos em diferentes camadas.

As classes e atributos do IFC são flexíveis e extensíveis, permitindo troca de informações entre fornecedores de *softwares* que atendem às diversas disciplinas necessárias a edificações em todo seu ciclo de vida. Segundo Eastman *et al.* (2011), as ferramentas de extração e inserção de informações devem ser customizáveis para trocar apenas as informações relevantes entre os sistemas. Entretanto, os mesmos autores afirmam que a flexibilidade do IFC, realizada por meio de atributos definidos pelo usuário (*userdefined*) ou indefinido (*notdefined*), permitindo que fornecedores de *software* e pesquisadores programem os tradutores para incluir informações adicionais de áreas não contempladas no IFC relacionadas a um objeto é um ponto de atenção. Contudo, pode-se adicionar informações somente à classe ou entidade existente no IFC.

Dentro de um fluxo de informações para garantir uma troca segura e efetiva, deve-se restringir informações pertinentes às necessidades de geração e consumo de cada interoperação ou troca de informações. Para que isso ocorra a buildingSMART desenvolveu o conceito de *Model Viewer Definition* (MVD).

O MVD permite selecionar apenas uma amostra de informações necessária para cada troca. A versão do IFC amplamente utilizada e mais proeminente é a 2x3. Nesta versão o MVD contém *templates* para visualização 3D e análises, tais como: interferências físicas, regras normativas, construtibilidade, análise estrutural, entre outras sobre demanda.

A buildingSMART lançou a versão 2x4 em 2012 que encontra-se em etapa de certificação das soluções por *software* para este formato. Nesta versão simplificada para IFC4, o MVD permite trabalhar com amostras ou subconjuntos duplicados em prol da restrição ao acesso da informação, contendo dois MVDs, sendo um com perfil público para interligar fluxos de trabalho e outro com perfil privado pertinente somente à operadora, cliente ou consórcio em que realiza-se o projeto do empreendimento.

Devido ao formato ser aberto e bem documentado, Amorim e Kassem (2015) afirmam que os governos da Finlândia, Noruega, Dinamarca e, principalmente, o Reino Unido têm exigido o IFC e o BIM como padrão de entrega em contratos para interligar fluxos de trabalho entre setor público e privado.

3.5. TRABALHOS RELACIONADOS AO IFC

O IFC é amplamente utilizado na exportação e importação de informações, em projetos da indústria na Arquitetura Engenharia, Construção e Operação (AECO), com mais de 150 softwares aderentes ao padrão (Abanda et al., 2015). Nesse contexto, nota-se que as ferramentas para extração e inserção de informações são fundamentais para promover a interoperabilidade através do padrão de arquivo IFC.

O conceito de Open BIM, segundo Chen e Luo (2014), refere-se ao método de gerenciamento de dados e modelo independente de *software*, formato específico ou proprietário. Os dados gerados em ferramentas proprietárias são exportados para formato de arquivo neutro, que torna a informação independente do *software* adotado. Além do IFC, existem outros formatos neutros utilizados no contexto de BIM para troca de informações entre sistemas e projetos. Entretanto o IFC é o padrão de dados Open BIM divulgado pela *International Alliance for Interoperability* (IAI), com base na experiência do padrão STEP. A ISO 16739 (2013) encontra-se próxima de uma utilização em larga escala em diversos países (Abanda *et al.*, 2015).

A utilização de ferramentas que atendem às funcionalidades BIM pode gerar ganhos relativos à produtividade. Sacks *et al.* (2010) comparou a produção de pré-moldados arquitetônicos para fachadas no modo tradicional com CAD e com o uso de ferramentas BIM. Mesmo com a falta de classes definidas para pré-moldados no padrão IFC, gerando uma perda em tradução dos objetos na troca de informações entre arquitetos e engenheiros estruturais, foi comprovado que a utilização do padrão demonstrou ganho em produtividade de 57% sobre o processo anterior e/ou tradicional realizado com ferramentas CAD 2D.

A extensão do IFC para que atenda às necessidades de outras áreas é encontrada frequentemente na literatura (Zhiliang *et al.*, 2011; Hegemann *et al.*, 2012; Jiao *et al.*, 2012; Janosch e Meschke, 2013; Zhang e El-Gohary, 2014, 2016; Beetz *et al.*, 2015; Motamedi *et al.*, 2016). Cheng e Das (2014) relatam que o IFC não contém uma estrutura de dados dedicada especificamente para eficiência energética e projeto de construção sustentável. Áreas qualitativas como o conforto humano ou referentes ao redor do ambiente construído não são atendidas pelo padrão IFC. Os autores afirmam que em pesquisas futuras o IFC pode ser estendido, proporcionando uma melhor abordagem ao projeto consciente de energia e gerenciamento dos dados de todo seu ciclo de vida.

É notório que o setor de óleo e gás convive com um sério problema causado pela falta de interoperabilidade entre sistemas de engenharia (Gallaher *et al.*, 2004; Pereira *et al.*, 2011). Existem várias iniciativas que propõem uma solução para este problema, como a norma ISO 15926 - *Standard for data integration, sharing, exchange, and hand-over between computer systems* (2004). Apesar deste padrão propor um nível de interoperabilidade acima do IFC (interoperabilidade semântica), este não possui maturidade para ser plenamente utilizado nas soluções por *software*. Para utilizar o IFC em projetos de tubulações para plantas industriais, faz-se necessário mapear os problemas encontrados na sua utilização em projetos da indústria AECO e registrá-los como lição aprendida. Diversas pesquisas relatam que o IFC precisa ser aprimorado para o uso de tecnologias de *web* semântica, atingindo a interoperabilidade semântica (Grilo e Jardim-Goncalves, 2010; Pauwels *et al.*, 2011; Irizarry *et al.*, 2013).

Sobretudo, outros pesquisadores relatam que o IFC necessita de novas classes e atributos para interligar informações entre funcionalidades BIM e o *Geographic Information System* (GIS), incorporando propriedades de pré-moldados e produtos temporários com foco em análises logísticas na construção para avaliar a viabilidade econômica (Irizarry *et al.*, 2013; Sacks *et al.*, 2010; Zhiliang *et al.*, 2011).

3.6. FERRAMENTAS E LINGUAGENS EXPLORADAS

Com o objetivo de declarar as ferramentas e formatos com linguagens padronizadas, ilustram-se as diferentes tecnologias exploradas nesta dissertação com suas respectivas definições:

• AVEVA Vantage PDMS (Plant Design Management System)

O AVEVA *Vantage* PDMS é um sistema que integra disciplinas utilizando um banco de dados central de estrutura própria associado a um programa CAD 3D. Este banco de dados contém arquivos de cada disciplina em disco gerenciados de forma unificada constituindo todo o projeto no CAD 3D. É possível acessar diversos módulos, tais como, de catálogos, especificações técnicas, geração de desenhos e isométricos.

Autodesk Software REVIT

O Revit permite a criação do modelo físico já que trabalha em um ambiente para modelagem 3D, mas esse programa não é capaz de realizar uma análise estrutural diretamente do modelo físico, permite importar e exportar IFC.

• Nemetschek SCIA Engineering

O SCIA Engineering é um software único que integra todas as fases do processo de projeto de uma estrutura: modelagem, análise, dimensionamento, detalhamento e desenhos. Realiza importação e exportação para diversos formatos, inclusive o IFC e a modelagem é realizada de forma paramétrica.

SAP 2000

A sigla S.A.P. é a abreviação de *Structural Analysis Program* e pertence à família de *softwares* para estruturas mais usada no mundo. O *software* SAP 2000 permite realizar análise estrutural, onde se efetuam os cálculos e projeta-se uma estrutura na aplicação fornecida.

Autodesk Software Robot

A ferramenta realiza análise e simulação de construções para estruturas, de forma precisa e aderente aos padrões requeridos. Esta ferramenta importa e exporta o formato IFC 2x3.

• BIM Vision

BIM Vision é um visualizador de modelo IFC. O BIM Vision visualiza os modelos BIM criados no formato IFC 2×3. Ele tem muitos recursos internos e é o primeiro visualizador com interface que permite usuários realizar *plug-ins* para adaptar as suas necessidades.

• Nemercheck Solibri Model Viewer

O Solibri Model Viewer (SMV) é gratuito e permite que você abra todos os formatos de arquivo IFC padrão. Trata-se de um visualizador 3D - Open BIM, que importa e exporta o formato de arquivo IFC, permitindo analisar modelos provenientes do CAD 3D. Essa análise do modelo pode ser relativa à qualidade, integralidade e de planejamento para construção.

A partir daqui são tratadas as extensões e linguagens dos arquivos, realizando sua descrição para declaração dos formatos utilizados na presente dissertação:

• eXtensible Markup Language (XML)

O XML é simplesmente uma sintaxe de documentos robusta e flexível. Nos últimos anos, o XML tem sido adotado em diversas áreas, justiça, finanças, hospitais, telecomunicações, agricultura, jornalismo, entre outras.

• JavaScript Object Notation (JSON)

O JSON é uma estrutura de dados em Java script que pode substituir o XML. A estrutura de dados fica mais simples de trabalhar e o tempo de execução de um *script* lendo dados em JSON é mais rápido do que ler um conteúdo XML.

• Steel Detailing Neutral Format (SDNF)

O SDNF é um formato padrão para a troca de informações de elementos de aços. Ele oferece um método neutro para a importação e exportação de dados do modelo estrutural. Este processo bidirecional permite a importação e exportação de arquivos SDNF de modelos estruturais.

• Datal (DTL)

O datal é um formato de arquivo texto nativo do PDMS, que funciona como um ponteiro para o catálogo de componentes e permite armazenar em tamanho pequeno projetos modelados no PDMS.

A proposta deste trabalho é utilizar o IFC 2x3 para interoperar informações necessárias aos projetos de tubulações industriais na área de óleo e gás, padronizando informações mínimas necessárias para projetar suportes e analisar a flexibilidade, dentro de um fluxo trabalho proposto no capítulo seguinte com implementações computacionais para otimizar o projeto de tubulações em plantas industriais.

4. PROPOSIÇÃO DE FLUXOS DE TRABALHO PARA AUMENTO DA MATURIDADE

Mediante ao que foi constatado no capítulo 2, a liberação imatura das tubulações para fabricação e montagem em empreendimentos de óleo e gás é uma das mais frequentes causas relatadas pelo grupo focal, gerando falhas de projeto e retrabalhos em virtude da desconexão dos fluxos de trabalho. Desta forma é fundamental o tratamento metodológico que garanta a maturidade deste processo. Assim é proposto um fluxo de trabalho que utiliza a interoperabilidade sintática, buscando otimizar e aumentar a maturidade em projetos de tubulação centrados no modelo 3D.

Este fluxo de trabalho é composto por um modelo sequencial com requisitos para concluir cada nível de maturidade proposto (seção 4.1), um processo colaborativo que envolve as competências dos projetistas e coordenadores na utilização das tecnologias (seção 4.2) e um fluxo de informações contendo os pacotes de dados gerados e consumidos na metodologia de maturidade proposta (seção 4.3), envolvendo processos, pessoas e tecnologias para melhoria do projeto executivo entregue à fabricação e montagem. Na seção seguinte propõe-se o modelo de maturidade.

4.1 MODELO DE MATURIDADE

A fim de garantir a maturidade necessária no processo de liberação das linhas de tubulação para fabricação, é apresentado um diagrama metodológico (Figura 14) que objetiva contribuir para a conformidade e otimização dos processos de engenharia desde a concepção até o "as built" (conforme construído) das tubulações industriais.

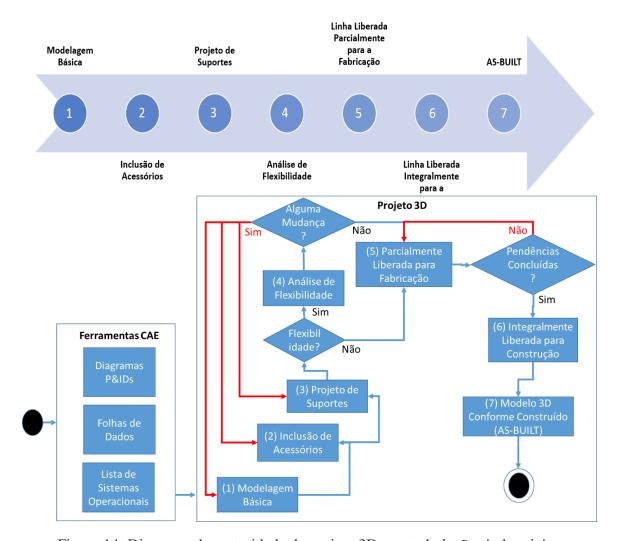


Figura 14: Diagrama de maturidade de projeto 3D para tubulações industriais.

Abaixo tem-se a descrição de cada passo apresentado no diagrama de maturidade (Figura 14):

(1) Modelagem Básica

Refere-se à modelagem do encaminhamento de uma linha de tubulação pelo projetista, desde a entrada até a saída de um fluido (roteamento completo conforme fluxograma de engenharia do projeto executivo), sem interrupções e sem acessórios. Entende-se que as informações provenientes da ferramenta CAE 2D são insumos automáticos de informações necessárias ao início da modelagem básica (fluxogramas de engenharia e dados de processo petroquímico consolidados). O coordenador técnico deve analisar e validar a qualidade do arranjo de cada linha de tubulação modelada.

(2) Inclusão de Acessórios

Refere-se à inclusão pelo projetista de todos os componentes de tubulação (incluindo drenos, figuras 8, instrumentos e demais acessórios). Nesta etapa todos os dados de engenharia aplicáveis devem estar preenchidos (conforme a taxonomia proposta), contendo os requisitos mínimos para a análise de flexibilidade se aplicável.

(3) Projeto de Suportes

O objetivo dessa etapa é projetar os suportes para uma linha de tubulação. Sendo assim, definem-se o perfil, material, ponto de aplicação, condições de contorno, carga e restrições de movimento. A partir desses aspectos definidos, pode-se realizar a análise estrutural e concluir o projeto do suporte, retornando o resultado da análise ao projeto 3D.

(4) Análise de Flexibilidade

A análise de flexibilidade tem o objetivo de reduzir os níveis das tensões secundárias, aumentando flexibilidade de uma linha de tubulação. Nesta etapa do modelo de maturidade, caso a linha de tubulação seja classificada pelo projetista mecânico como passível de análise, deve-se realizar o cálculo de flexibilidade. Caso exista a necessidade de mudança no arranjo da linha de tubulação ou no projeto de suportes, retorna-se à informação para retrabalhar as etapas 1, 2 ou 3 do projeto 3D, realizando um fluxo de trabalho padronizado centrado no modelo de maturidade proposto.

(5) Análise de Impeditivos para Fabricação (Linha Liberada Parcialmente para a Fabricação)

Linha liberada parcialmente para fabricação e montagem com todos os suportes e sem interferências físicas, independentemente de sua criticidade. Entende-se por linha parcialmente liberada aquela que contém um ou mais trechos liberados para construção, desde que os referidos trechos liberados não sejam passíveis de modificações. A suportação da linha deve estar consolidada, independentemente de sua criticidade (etapa 3). Se houver análise de flexibilidade da linha, a mesma deve estar aprovada (etapa 4). Nesta etapa são monitoradas e analisadas as pendências que impedem cada linha de tubulação de ser liberada integralmente para a fabricação.

(6) Linha Liberada Integralmente para a Fabricação

Todas as pendências impeditivas e não impeditivas devem ser eliminadas para aprovação da linha de tubulação. Sendo assim, não é permitida a existência de componentes em *HOLD* (espera, ou seja, aguardando definições para consolidação), devendo estar com total integridade da linha de tubulação para ser liberada à fabricação e montagem.

(7) As Built (Conforme Construído)

Nesta etapa de adequação da modelagem conforme o construído, deve-se analisar em campo cada linha de tubulação e atualizá-las no projeto 3D. Desta maneira, deve ser verificado no projeto se todos os itens se encontram conforme o construído. Trata-se de uma fase importante para a operação e manutenção do ativo, pois qualquer análise ou alteração do projeto deve ser precisa, do contrário pode-se ocasionar erros e descrédito sobre o modelo 3D. Sobretudo, o objetivo desta atividade é proporcionar a credibilidade do projeto 3D para democratização das informações durante seu ciclo de vida.

Na seção seguinte é apresentado o processo colaborativo proposto, buscando maior sinergia entre colaboradores e ferramentas de automação de projetos, dentro do fluxo de trabalho de maturidade proposto para entrega dos projetos de tubulação às etapas subsequentes do processo, tais como, suprimento, fabricação, construção e montagem, comissionamento, operação e manutenção.

4.2 PROCESSO COLABORATIVO

Neste modelo proposto, os colaboradores envolvidos no projeto de tubulações industriais são projetistas que modelam as linhas de tubulação na ferramenta CAD/CAE 3D, conforme determinado pelos fluxogramas de engenharia. Os projetistas mecânicos especificam os materiais no catálogo e realizam análises de flexibilidade em ferramentas específicas, a partir de um formato de arquivo neutro e padronizado. Os projetistas estruturais específicam suportes de tubulação em ferramentas Open BIM ou CAD/CAE 3D, confirmam o perfil e material mais adequado para executar a análise estrutural interoperando entre ferramentas Open BIM e ferramenta CAD/CAE 3D.

O engenheiro responsável pelo projeto coordena o processo até a aprovação de cada linha de tubulação. A equipe de projeto executivo pode ser dimensionada de acordo com a quantidade, tempo e pacote de serviço planejado, atendendo a data de necessidade requerida

pela fabricação e montagem. Desta forma, permite-se iniciar seus fluxos de trabalho, conforme orquestração da construtibilidade (habilidade de analisar aspectos construtivos previamente à construção), planejamento e controle para atingir objetivos estratégicos.

O planejamento e controle neste processo é responsável por programar e sinalizar quando ocorre a liberação de cada linha de tubulação com sua respectiva prioridade no sequenciamento de montagem. A partir disso, os planejadores exportam da ferramenta CAD/CAE 3D quantitativos de materiais, status do projeto, estimativas e determinam as equipes necessárias, identificando oportunidades e prioridades para execução. Vale a pena salientar que se deve programar as atividades com decisão em consenso interdisciplinar. Os colaboradores recebem o projeto consolidado de tubulações industriais e são os responsáveis pelo suprimento, fabricação, montagem, segurança, comissionamento e qualidade de um empreendimento.

Estes colaboradores devem interagir desde a concepção do projeto por meio da gestão visual 3D colaborativa e podem pós-processar isométricos, lista com dados de engenharia, lista de materiais, lista de soldas, lista de *spools* (trechos de uma linha de tubulação para facilitar a logística e montagem), lista de sistemas operacionais, lista de áreas físicas de construtibilidade e determinar em consenso prioridades estipulando datas de necessidade para conclusão de cada área física e sistemas operacionais. As inconsistências nas informações fornecidas pela ferramenta CAD/CAE 3D podem ser verificadas na fase de projeto executivo, com o mínimo de intervenção humana na troca de dados entre sistemas. Para otimizar duas etapas do modelo de maturidade (3 e 4) é apresentado na seção seguinte um fluxo de informações proposto para projetar tubulações utilizando a interoperabilidade.

4.3 FLUXOS DE INFORMAÇÕES

Nesta seção apresenta-se um fluxo de informações, visando aumentar a maturidade e facilitar o processo colaborativo dos projetos de tubulações industriais, com soluções computacionais. Esse fluxo de processos preconiza o cumprimento de cada etapa antes da liberação de uma linha para fabricação, priorizando o uso da interoperabilidade sintática para otimizar a troca de informações entre sistemas em projeto de suportes (etapa 3) e análise de flexibilidade das tubulações (etapa 4). A figura 15 ilustra o fluxo de informação proposto.

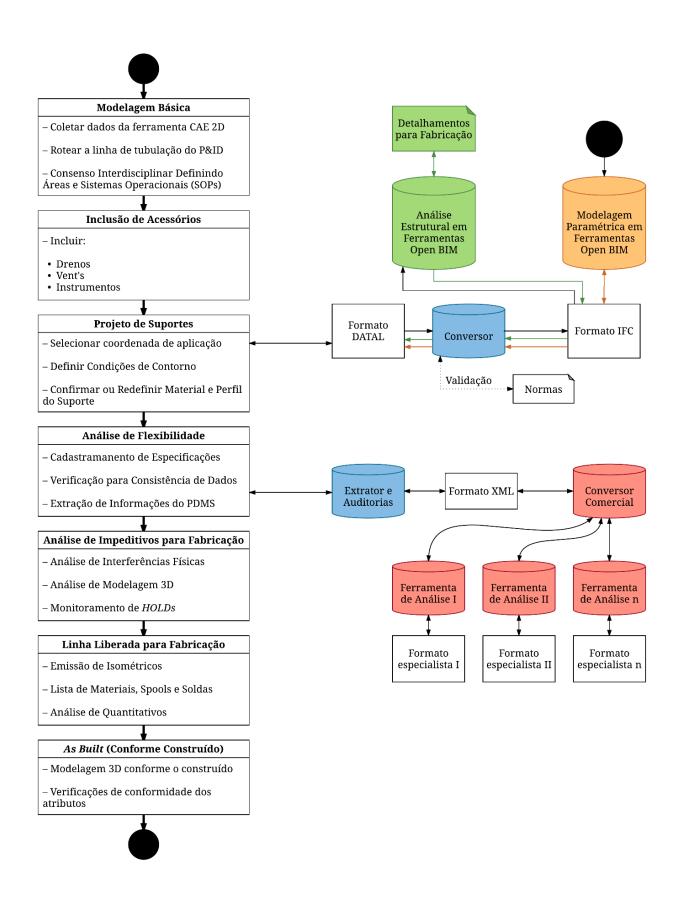


Figura 15: Fluxo de Informações para projeto 3D de tubulações industriais.

Na etapa 3, no fluxo de informações explora-se a interoperação para definir qual suporte será utilizado, trocando informações entre suportes modelados em ferramentas Open BIM ou modelados na ferramenta CAD/CAE 3D. Para isso, utiliza-se o formato Datal para troca de dados bidirecional com o formato IFC, conectando as informações exportadas do PDMS com ferramentas Open BIM. Nesse fluxo de trabalho é proposto o detalhamento dos perfis (suportes estruturais) em ferramentas Open BIM de análise estrutural e retorno para ferramenta de modelagem por meio do IFC. O processo para elaboração do projeto 3D da suportação em tubulações industriais permite atuar em duas hipóteses:

a) supondo que a empresa não possua catálogo de suporte em PDMS, o projetista estrutural modela os suportes em ferramenta Open BIM, utilizando o conversor desenvolvido e gerando o formato IFC no formato Datal. A partir disso, o arquivo Datal gerado deve ser importado no PDMS, determinando-se a coordenada e condições de contorno do suporte. Para exportação do arquivo Datal com os resultados das definições, há a necessidade da utilização do conversor para a transformação do Datal para o IFC. De posse do arquivo, realiza-se importação na ferramenta de análise estrutural Open BIM selecionada. Por fim, utiliza-se novamente o conversor para retornar à definição do suporte definitivo no PDMS com resultado da análise estrutural, confirmando o perfil e material mais adequado no projeto.

b) supondo que a empresa possua catálogo de suportes no PDMS, o projetista estrutural exporta arquivo Datal da linha de tubulação contendo os suportes para transformar no formato IFC com o conversor desenvolvido. Após a transformação, o arquivo IFC deve ser importado na ferramenta Open BIM selecionada para análise estrutural. Utiliza-se o conversor para transformar o resultado da análise para o formato Datal, importando e atualizando o PDMS, confirmando informações definidas acerca do perfil e materiais consolidados para cada suporte.

Nas duas hipóteses de processo deve ser definida obrigatoriamente uma norma, podendo ser importada por uma tabela no conversor desenvolvido. Dentro do processo de projeto da suportação, uma das causas identificadas com maior ocorrência é a falta de detalhamento do projeto para a fabricação. Nesse contexto essa etapa deve utilizar soluções por *software* que realizam dimensionamento e detalhamento das estruturas, após análise e definições dos perfis que serão utilizados. Ao final é necessário exportar o arquivo IFC com os resultados e utilizar visualizadores Open BIM para criar procedimentos destacando as armaduras ou ligações metálicas em documento a ser enviado para a fabricação, além disso, deve-se retornar à informação para ferramenta de modelagem Open BIM.

Na etapa 4 o fluxo de informações explora a extração e conversão das informações necessárias à análise de flexibilidade no PDMS, extraindo e verificando se as informações mínimas necessárias estão preenchidas no projeto 3D até atingir a maturidade necessária para realização do cálculo de flexibilidade. Nesta etapa, por não existir classes e atributos dedicados no IFC para tubulações industriais e análise de flexibilidade, utilizou-se o XML por ser uma linguagem comumente usada em tradutores.

Desta forma, o projetista mecânico utiliza extrator de dados desenvolvido no PDMS e transforma em um formato neutro de arquivo XML. Este extrator desenvolvido também verifica internamente no PDMS o preenchimento de cada atributo necessário à análise de flexibilidade no catálogo e no módulo *Design*.

Caso algum atributo não esteja preenchido, a função informa as pendências de preenchimento no banco de dados do PDMS. Deve-se somente exportar o arquivo XML se os atributos mínimos para análise estiverem preenchidos. O projetista mecânico selecionará previamente qual ferramenta é mais adequada para a análise, levando em consideração a leitura do arquivo XML ou conversão para seu formato padrão de importação.

Após apresentar o modelo de maturidade proposto e seu respectivo fluxo de informações em projetos de tubulações industriais, descreve-se no capítulo seguinte os estudos empíricos computacionais, necessários para avaliar a troca de informações entre sistemas com os conversores desenvolvidos para projeto de suportes e análise de flexibilidade das tubulações, respectivamente, as etapas 3 e 4 do fluxo de informações.

5. ESTUDO EXPERIMENTAL

Neste capítulo são exploradas e descritas as implementações computacionais realizadas, contendo diagramas de arquiteturas das soluções e detalhamento das técnicas de desenvolvimento, assim como quais atributos e métodos foram utilizados. Desta maneira, detalham-se as funcionalidades das soluções por *software* desenvolvidas, explorando a interoperabilidade sintática na terceira e quarta etapas do modelo de maturidade proposto: projeto de suportes e análise de flexibilidade (vide figura 15).

Inicialmente são apresentadas as tecnologias e ferramentas Open BIM atuais ligadas ao escopo deste trabalho, em um experimento preliminar, anterior às implementações computacionais, para efeito de verificação e validação das tecnologias a serem aplicadas nos estudos empíricos. Para isso, na seção 5.1 é realizado teste de importação e exportação de suportes modelados no Autodesk Revit e em diversas ferramentas Open BIM.

Em seguida, para analisar em projetos reais de plantas industriais, verificando as vantagens e desvantagens da aplicação do fluxo de trabalho proposto e das tecnologias desenvolvidas, são descritos os estudos empíricos que utilizam a interoperabilidade sintática (seções 5.2, 5.3 e 5.4). Os dados foram coletados de projetos já realizados, verificando a conformidade dos suportes projetados perante a norma AISC (2010) e a consistência de dados mínimos necessários à análise de flexibilidade no projeto 3D.

A investigação empírica é realizada em dois projetos distintos de plantas industriais, extraindo informações legadas de projetos 3D, convertendo-as do formato proprietário da ferramenta de modelagem 3D para o IFC e importando-as em ferramentas de análise estrutural ou visualização 3D disponíveis.

Após realizar a análise estrutural dos suportes, retornam-se os resultados da análise ao projeto 3D (seções 5.2 e 5.3). Além disso, a terceira avaliação empírica desta pesquisa (seção 5.4), testa por meio de informações de uma planta de refino de petróleo a extração e consistência dos dados gerados do projeto 3D de tubulações para o formato neutro proposto, destinado a satisfazer os requerimentos para análise de flexibilidade. Ao final da seção 5.4, realiza-se uma análise crítica dos dados extraídos em prol da conformidade no preenchimento dos atributos mínimos que permitam a interoperação sintática de informações para o cálculo de flexibilidade.

5.1 TESTES DE INTEROPERABILIDADE ENTRE FERRAMENTAS OPEN BIM

A primeira ação realizada para avaliação empírica do processo proposto, trata-se da análise sobre a interoperabilidade sintática entre as seguintes soluções por *softwares* Open BIM: *Autodesk Revit*, *Autodesk Robot*, *Solibri Model Viewer* e *SAP 2000*. Para isso, modelou-se um conjunto de suportes no *Autodesk Revit 2016*. Dentre eles, encontram-se perfis metálicos e de concreto, como ilustra a Figura 16.

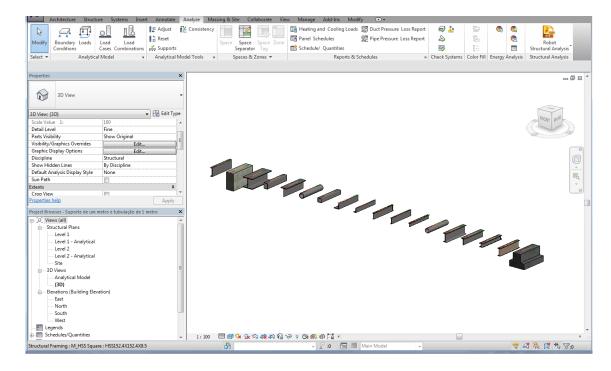


Figura 16: Conjunto de suportes modelados no Autodesk Revit.

A partir da modelagem paramétrica, exportou-se um arquivo *IFC 2x3* desses suportes do *Autodesk Revit 2016* para analisar no *Autodesk Robot* a efetividade de importação deste formato em uma ferramenta de análise da mesma fabricante de *software*, conforme ilustra a Figura 17.

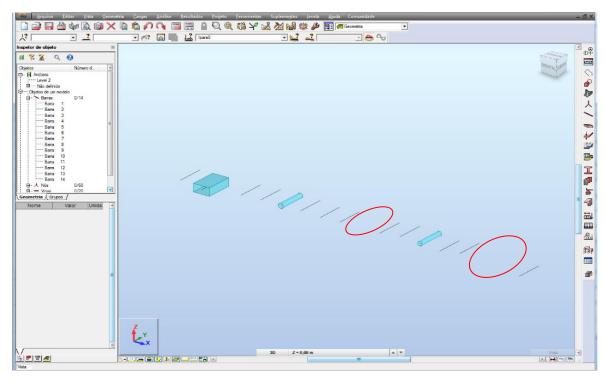


Figura 17: Conjunto de suportes importados no *Autodesk Robot* via *IFC 2x3*.

Nota-se que algumas inconsistências são identificadas na importação dos dados geométricos e não geométricos. Sobretudo, o *Autodesk Robot* não identificou alguns suportes circulados em vermelho como elementos de barra e em todos os casos não reconheceu as propriedades do material de cada suporte. Desta forma, realizaram-se testes em diferentes classes do *IFC 2x3* e em nenhum deles o problema da leitura nessa ferramenta foi solucionado.

Nesse contexto, foi também verificada a transferência de informações quando esta é feita através do módulo de integração direta existente entre o *Autodesk Robot e o Autodesk Revit* (canto superior direito da Figura 16). O resultado mostra que embora na maioria dos suportes as informações geométricas e não geométricas tenham sido preservadas, é possível perceber que ainda assim dois suportes são reconhecidos apenas como elementos de barra, conforme é demonstrado na Figura 18.

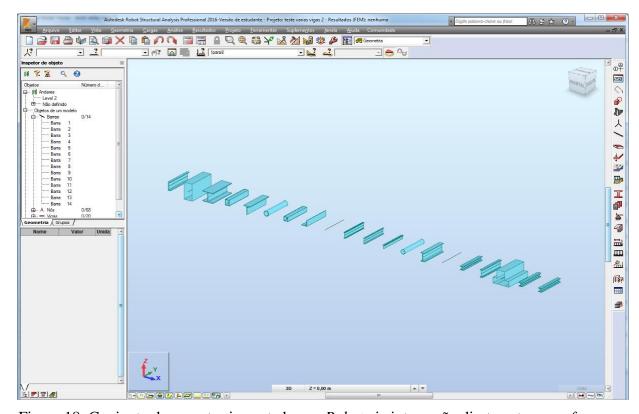


Figura 18: Conjunto de suportes importados no *Robot* via integração direta entre os *softwares*.

Mediante ao exposto, realizou-se uma análise da interoperabilidade sintática entre *Autodesk Revit* e *Autodesk Robot* utilizando o *IFC 2x3*. A Tabela 3 apresenta um relatório que enfatiza a verificação da importação através do *IFC 2x3*, onde é possível observar que a grande maioria dos perfis metálicos não foram importados pelo *Robot*, exibindo uma deficiência desta solução por *software* na leitura deste formato.

Tabela 3: Relatório do Autodesk Robot com análise dos suportes importados via IFC.

Família e Tipo	Material Estrutural	Existência	Seção	Material <i>Robot</i>
M_Precast-Inverted Tee: 600 IT 500	Concrete - Precast Concrete - 35 Mpa	Não	Não	Betão
UB-Universal Beams: UB254x102x28	Metal - Steel 43-275	Sim	Não	Betão
M_W Shapes: W130X28.1	Steel, 45-345	Não	Não	Betão
M_ST Shapes: ST230X52	Steel, 45-345	Sim	Não	Betão
M_S Shapes: S310X74	Steel, 45-345	Sim	Não	Betão
M_Pipe: Pipe152STD	Steel, 45-345	Sim	Sim	Betão
M_MT Shapes: MT125X6.7	Steel, 45-345	Sim	Não	Betão
M_MC Shapes: MC200X12.6	Steel, 45-345	Sim	Não	Betão
M_M Shapes: M200X9.7	Steel, 45-345	Não	Não	Betão
M_L Unequal Angles: L152X102X12.7	Steel, 45-345	Sim	Não	Betão
M_L Equal Angles: L203X203X12.7	Steel, 45-345	Sim	Não	Betão
M_HSS Square: HSS152.4X152.4X9.5	Steel, 45-345	Sim	Não	Betão
M_HSS Round: HSS190.5X12.7	Steel, 45-345	Sim	Sim	Betão
M_WT Shapes: WT305X62.5	Steel, 45-345	Sim	Não	Betão
M_HSS Rectangular: HSS203.2X152.4X9.5	Steel, 45-345	Sim	Não	Betão
M_HP Shapes: HP360X174	Steel, 45-345	Não	Não	Betão
M_Concrete-Rectangular Beam: 300 x 600mm	Concrete, Cast-in-Place gray	Sim	Sim	Betão
M_C Shapes: C310X45	Steel, 45-345	Sim	Não	Betão

Na Tabela 3, é possível identificar que o *Autodesk Robot* conseguiu ler as coordenadas de maioria dos suportes, porém não possibilitou representar a geometria de praticamente todos com exceção de três perfis destacados em cinza, sendo dois circulares de aço e um retangular de concreto. Pode-se perceber que a classe *IfcMaterial* (vide Tabela 3: coluna Material *Robot*) não é identificada pelo sistema, uma vez que todos os materiais são ignorados e substituídos pelo material Betão.

Na figura 18, via integração direta (Revit para Robot), ocorrem erros na leitura de apenas duas geometrias, utilizando o formato da *Autodesk*. Sobretudo, avaliando a importação do arquivo IFC é possível perceber que 17% das geometrias foram processadas e nenhum dos atributos mínimos foram identificados.

Após importação do arquivo *IFC 2x3* no *Autodesk Robot*, avalia-se a viabilidade de utilização da interoperabilidade sintática entre o *IFC 2x3* e o *software SAP 2000* para análise estrutural. A Figura 19 demonstra que todos os suportes são importados com efetividade, preservando dados geométricos e não geométricos (inclusive os materiais) necessários para a análise estrutural.

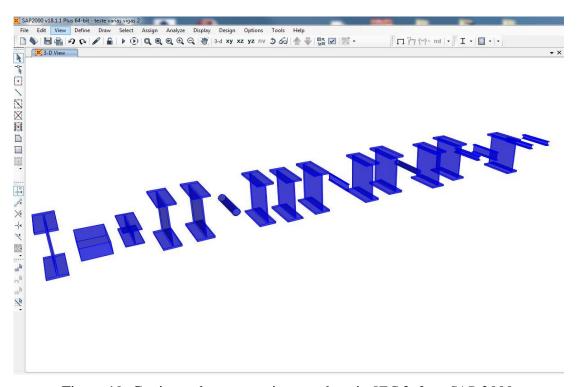


Figura 19: Conjunto de suportes importados via *IFC 2x3* no *SAP 2000*.

Com o objetivo de analisar se este arquivo *IFC 2x3* gerado a partir do *Autodesk Revit 2016* é importado com efetividade em visualizadores 3D Open BIM, realiza-se a leitura no *software Solibri Model Viewer* e *BIM Vision*, analisando se os dados geométricos e não geométricos dos suportes são preservados. O resultado da importação no *Solibri Model Viewer* exibido na Figura 20 demonstra que tanto os dados de engenharia quanto as geometrias são preservadas.

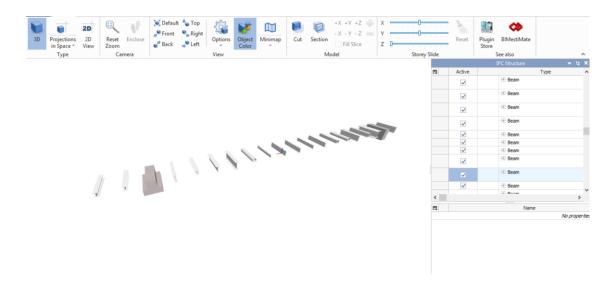


Figura 20: Conjunto de suportes no BIM Vision.

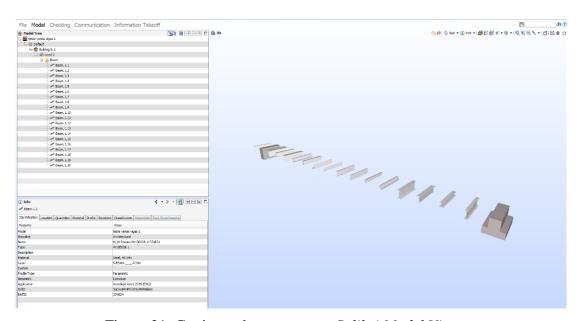


Figura 21: Conjunto de suportes no Solibri Model Viewer.

Sendo assim, para verificar se o erro concentra-se na importação do *Autodesk Revit* 2016, restou analisar se sua leitura preserva as informações geométricas e não geométricas importadas via *IFC* 2x3. Desta forma, é modelado um perfil metálico L64X6 no *Autodesk Revit* 2016 e exportado em arquivo *IFC* 2x3, conforme é exibido na Figura 22.

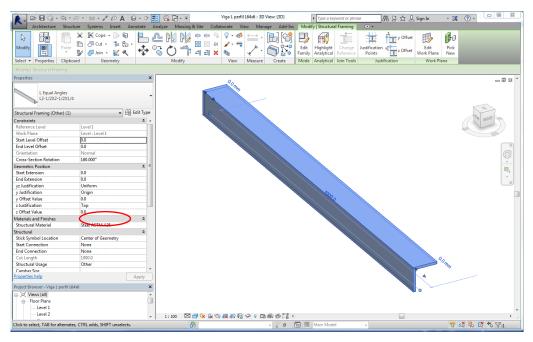


Figura 22: Suporte L64X6 criado no Revit, com material definido "Steel ASTM A36".

Conforme ilustra-se na Figura 23, o suporte é exportado para o formato IFC 2x3 Coordination View 2.0.

```
#204= IFCAXIS2PLACEMENT2D (#202,#23);

#205= IFCCIRCLE (#204,3.175);

#206= IFCTRIMMEDCURVE (#205, (IFCPARAMETERVALUE (0.)), (IFCPARAMETERVALUE (90.00000000001)),.T.,.PARAMETER.);
#205= IFCCIRCLE (#204, 3.175);
#206= IFCCIRCLE (#204, 3.175);
#206= IFCCIRCLE (#204, 3.175);
#206= IFCCIRCLE (#204, 3.175);
#206= IFCCIRCLE (#204, 3.175);
#209= IFCCOMPOSITECURVESCEMENT (.CONTINUOUS., T., #206);
#210= IFCCARTESIANPOINT ((-20.8138888888888, 39.5111111111124));
#212= IFCCARTESIANPOINT ((-23.98888888888, 39.5111111111155));
#214= IFCCARTESIANPOINT ((-23.988888888883, 39.5111111111155));
#215= IFCCARTESIANPOINT ((-23.988888888883, 39.5111111111125));
#216= IFCCARTESIANPOINT ((-23.988888888893, 39.5111111111125));
#217= IFCCARTESIANPOINT ((-23.98888888893, 39.5111111111125));
#218= IFCCARTESIANPOINT ((-23.98888888893, 39.5111111111125));
#221= IFCCARTESIANPOINT ((-23.98888888893, 39.5111111111125));
#222= IFCCOMPOSITECURVESCEMENT (.CONTINUOUS., T., #221);
#223= IFCCARDSITEARYCLOSEDPROFILEDEF (.AREA., 'L2-1/2X2-1/2X1/4', #224);
#233= IFCCARDSITEARYCLOSEDPROFILEDEF (.AREA., 'L2-1/2X2-1/2X1/4', #224);
#233= IFCCARTESIANPOINT ((0., 5.9294888888878, 39.511111111126));
#233= IFCCARTESIANPOINT ((0., 5.9294888888878, 39.511111111126));
#233= IFCCARTESIANPOINT ((0., 5.9294888888878, 39.5111111111126));
#233= IFCCARTESIANPOINT ((0., 5.9294888888878, 39.5111111111126));
#233= IFCCARTESIANPOINT ((0., 5.9294888888878, 39.5111111111126));
#233= IFCCARTESIANPOINT ((0., 5.92948888888888, 39.5111111111126));
#233= IFCCARTESIANPOINT ((0., 5.92948888888888, 39.51111111111126));
#244= IFCCOLOURING (6, 0.780392156862745, 0.780392156862745);
#244= IFCCARTESIANPOINT ((0., 6242));
#244= IFCCOLOURING (6, 0.780392156862745, 0.780392156862745);
#244= IFCCARTESIANPOINT ((1000., 0.));
#244= IFCCARTESIANPOINT ((1000., 0.));
#254= IFCSHAPEREFESENTATION (#98, 'Body', 'SweptSolid', (#239));
#255= IFCCARTESIANPOINT ((1000., 0.));
#256= IFCSHAPEREFEREERINT ((6, 0.7803));
#257= IFCCARTESIANPOINT ((1000., 0.));
#258= IFCMATERIAL ('Sceel ASTM A36');
#277= IFCSTYLEDITEM(8, (#275), $);
#278= IFCSTYLEDITEM(8, (#275), $);
#279= IFCSTYLEDITEM(8, (#275), $);
#279= IFCSTYLEDITEM(8, (#275), $);
#279= IFCSTYLEDITEM(8, (#275), $);
#279= IFCSTYLED
```

Figura 23: Arquivo IFC 2x3 com o suporte L64X6 criado no *Revit*.

De posse do arquivo IFC 2x3 demonstrado na Figura 23, é avaliado por meio de uma importação no Autodesk Revit 2016 a preservação das informações. É possível notar, por exemplo, que na Figura 24 novamente perdeu-se a informação da classe *IfcMaterial*. Entretanto, os dados geométricos foram preservados.

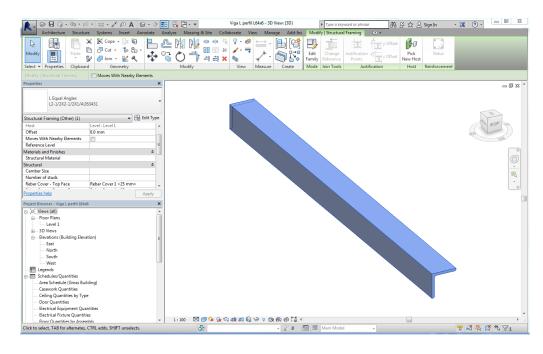


Figura 24: Suporte L64X6 criado no Autodesk Revit, sem informações de materiais.

Portanto, o erro na leitura das informações contidas no IFC é comprovado com o exemplo da classe *IfcMaterial*. Dentro desse contexto, com o objetivo de adquirir conhecimento necessário para implementação de uma solução por *software* de leitura e escrita do *IFC 2x3* ou o *Datal*, realiza-se manualmente a escrita de um arquivo *IFC 2x3* no formato *Json (JavaScript Object Notation)*, a partir da seleção de um suporte analisado nessa seção entre ferramentas *Open BIM*, para estimular o entendimento da estrutura de dados apresentada pelo *IFC*, conforme demonstrado na Figura 25.

Figura 25: Escrita de um formato Json a partir do IFC 2x3 utilizado nos testes.

A partir do arquivo em formato *Json* é possível entender o relacionamento das classes e atributos de forma sequencial, especificamente para avaliar os tipos de perfis e suas características geométricas ou não geométricas. Sendo assim, desenvolveu-se para facilitar a leitura de diferentes arquivos *IFC 2x3* um conversor na linguagem C++ que gera arquivos no formato *Json*. A Figura 26 ilustra uma conversão.

Figura 26: Conversão para o formato *Json* a partir do *IFC 2x3* utilizado nos testes.

É possível afirmar que os estudos realizados neste tópico puderam apoiar o processo de aprendizado de hierarquia das entidades ou classes para facilitar a implementação. Além disso, avaliar quais ferramentas estão aptas a serem utilizadas para validar os conversores desenvolvidos com dados reais de projetos de instalações industriais. Na seção seguinte é descrita a solução computacional para a etapa três do modelo proposto, atingindo-se a interoperabilidade sintática em especificar, converter e calcular as estruturas de suportação utilizando o *IFC*.

5.2. DESENVOLVIMENTO DE CONVERSOR BIDIRECIONAL PARA PROJETO DE SUPORTES

Realizam-se estudos no *software Plant Design Management System* (PDMS) em busca do entendimento de qual arquivo pode ser gerado e quais informações mínimas necessárias devem ser extraídas para análise estrutural. Em paralelo foram realizados diversos testes de importação e exportação em ferramentas Open BIM, avaliadas na seção anterior. Nesta etapa, foram consultados especialistas no PDMS e estruturas civil. Estes especialistas comentaram quais informações são minimamente necessárias e que limitações deveriam ser destacadas como pontos de atenção.

Nesse contexto, selecionou-se o formato de arquivo *Steel Detailing Neutral File* (SDNF), realizando diversos testes a partir de sua exportação do PDMS para compatibilidade com o formato IFC. A partir deste arquivo é possível notar que as informações do catálogo de suportes são referenciadas apenas por uma chave primária e poucas características dos objetos estavam escritas no arquivo. Outro aspecto percebido após diversos testes realizados com esse formato tratou-se da quantidade de transformações que seriam geradas na interoperabilidade entre PDMS e ferramentas Open BIM, atingindo em alguns casos quatro transformações, o dobro se comparado com a utilização do Datal para conversão bidirecional.

Conforme explicitado na teoria no capítulo 3, a diferença entre os conceitos de integração ponto-a-ponto (P2P) e interoperabilidade sintática, difere-se na quantidade de transformações para interligar as ferramentas necessárias nos estudos experimentais desta dissertação, que se concentra em projetos de tubulações industriais, conforme é exemplificado na Figura 27.



Figura 27: Integração ponto-a-ponto à esquerda e interoperabilidade sintática com IFC à direita.

Pode-se observar que a integração entre sistemas, especificamente nas ferramentas envolvidas nos experimentos, sem respeitar os requisitos da interoperabilidade, pode aumentar o número de transformações. Neste estudo, faz-se necessário 23 transformações para integrar informações entre todos os *softwares* utilizados. Contudo, a utilização da interoperabilidade sintática, por meio de um formato com linguagem padronizada, reduz a quantidade de transformações para 5 interoperações e torna viável a livre escolha por requisitos técnicos das ferramentas no segmento de montagem industrial.

De acordo com essa constatação, foi alterada a estratégia de testes para implementação. A partir dessa barreira ou alteração, iniciaram-se estudos no formato nativo do PDMS (Datal) para entendimento de sua estrutura de dados e realização dos testes de exportação e importação. Esses testes levantaram a necessidade da especificação de informações mínimas necessárias para interoperar informações com ferramentas de análise estrutural.

Após essa especificação e alcance da etapa de compatibilização com IFC, analisando cada classe geométrica e dados de engenharias (propriedades), percebeu-se que o formato nativo do PDMS não descrevia todas as informações requeridas pelo IFC. Na investigação das causas, foi possível identificar que o arquivo Datal não especificava cada aresta pertencente à viga ou suporte como no IFC, conforme ilustra Figura 28.

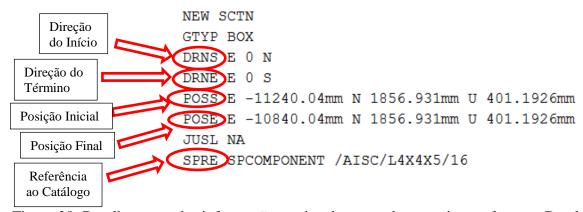


Figura 28: Detalhamento das informações mais relevantes de uma viga no formato Datal.

Desta forma, adotou-se uma estratégia de compatibilização das informações dimensionais à mesma norma, neste caso AISC (2010). Após essa compatibilização com norma internacional, uma lógica foi detalhada para converter do formato IFC para o Datal. Primeiramente foi priorizado a conversão do IFC para o Datal. O Datal necessita das coordenadas iniciais e finais de uma viga, o plano de corte, a posição da linha neutra e o apontamento do tipo de viga para o catálogo do PDMS.

Sendo assim, deixou-se o plano de corte padrão, ou seja, sem inclinações e manteve-se a linha neutra sempre no meio da seção da viga (centroide geométrico). Portanto, foi preciso extrair do IFC a coordenada inicial, a direção e o comprimento da viga para poder representála no formato de arquivo Datal. Sobretudo, foi necessário capturar o nome da viga no catálogo do IFC, verificar suas dimensões na tabela da norma AISC (2010) e fazer uma tradução para a viga correspondente no PDMS.

Com o tratamento por palavras-chave foi possível implementar a busca e automatizar a conversão coletando as informações necessárias direto da norma internacional AISC (2010), assim garantindo a qualidade na interoperabilidade sintática entre sistemas de automação de projetos e ferramentas *Open BIM* por meio do *IFC* e não apenas uma integração ponto a ponto entre ferramentas de automação de projetos.

Para interligar o projeto de tubulação com a suportação dentro de um fluxo de trabalho em prol da conclusão do projeto executivo, é proposto um conversor bidirecional capaz de ler e escrever os formatos *Datal* e *IFC* respectivamente. Os suportes podem ser criados no *PDMS* e exportados no formato Datal ou criados em uma ferramenta *Open BIM*, sendo exportados no formato *IFC*. Portanto, esse conversor é capaz de realizar esta interoperação bidirecionalmente entre as diferentes ferramentas utilizadas, compatibilizando as informações com normas aplicáveis entre os sistemas de engenharia envolvidos.

O conversor desenvolvido é capaz de ler um arquivo texto, identificar se o formato é datal ou IFC e fazer a conversão necessária. Para realizar o desenvolvimento desse conversor, foi adotada a linguagem de programação *Python*. Esta linguagem foi escolhida por possuir recursos em sua biblioteca padrão além de módulos desenvolvidos por terceiros.

Nesse trabalho, utilizou-se a biblioteca *IfcOpenShell-python* 2.7 32bit para *Windows*, uma biblioteca de código aberto que auxilia usuários e desenvolvedores de *softwares* a trabalhar com o *IFC*, formato utilizado mais comumente na metodologia *BIM*. Essa biblioteca possibilitou tanto ler e interpretar o arquivo *IFC* quanto criar classes para um novo *IFC*, tornando-se fundamental para a conclusão do conversor bidirecional. O *PyCharm*, desenvolvido pela *JetBrains*, foi utilizado como ambiente de desenvolvimento, já que foi criado especialmente para a linguagem *Python*. A Figura 29 ilustra um fluxograma com as principais etapas para funcionamento dos conversores.

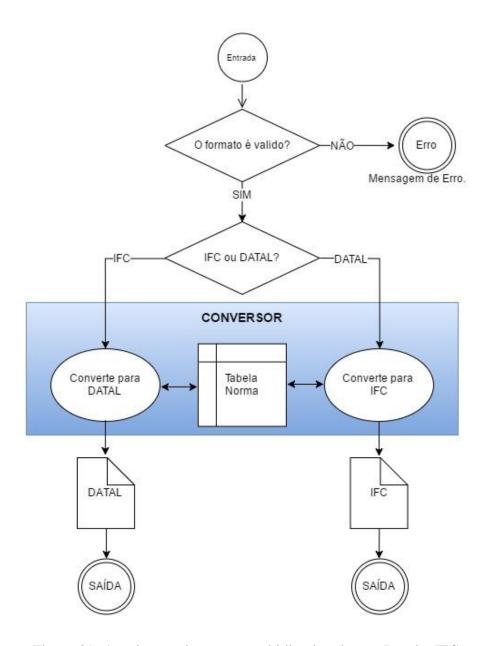


Figura 29: Arquitetura do conversor bidirecional entre Datal e IFC.

O conversor é capaz de receber um determinado arquivo e verificar se este é válido para a conversão. A solução por *software* realiza a conversão necessária para cada caso, pois é capaz de identificar se o arquivo é IFC ou Datal. A conversão do IFC para Datal consiste em ler a geometria dos elementos estruturais filtrando apenas as informações necessárias para análise estrutural. Esse filtro é chamado de *Model View Definition* (MVD). Trata-se de um arquivo que contém todas informações requisitadas para realizar uma simulação ou análise desprezando tudo o que não é necessário para a ferramenta onde será imputado.

Ao verificar detalhadamente o padrão do Datal, nota-se que seu formato contém informações limitadas, pois não possui todas as informações necessárias para gerar a geometria de um perfil metálico, necessitando assim de compatibilização das dimensões com uma norma.

Da mesma forma, quando o arquivo de entrada é um *Datal*, o conversor identifica os elementos estruturais dentro do arquivo e coleta o tipo de perfil através de sua nomenclatura. Em seguida, um arquivo de saída no formato *IFC* é gerado. Vale ressaltar que para o pleno funcionamento do *Datal* no *PDMS*, deve-se ter o perfil cadastrado no catálogo e sua chave primária deve ser padronizada e compatível com a norma aplicada. O nome do perfil metálico que está entre as informações contidas no *IFC* é facilmente encontrado dentro da norma, e todos os parâmetros necessários para a geração do Datal são coletados.

Nesta versão, o conversor é capaz de interpretar apenas a norma americana AISC (2010), mas uma expansão para as demais normas pode ser facilmente acrescentada, compatibilizando os atributos necessários em uma tabela. O conversor permite a importação de tabelas contendo as informações dimensionais minimamente correlatas a AISC (2010).

Assim como no processo de leitura do arquivo no padrão IFC apresentado no capítulo 4, a implementação do conversor de Datal para IFC inicialmente vai buscar a entidade da camada mais externa definido pela hierarquia de entidades para começar a conversão. Para exemplificação seguem os passos para a conversão de uma viga a partir do trecho de código presente na Figura 30.

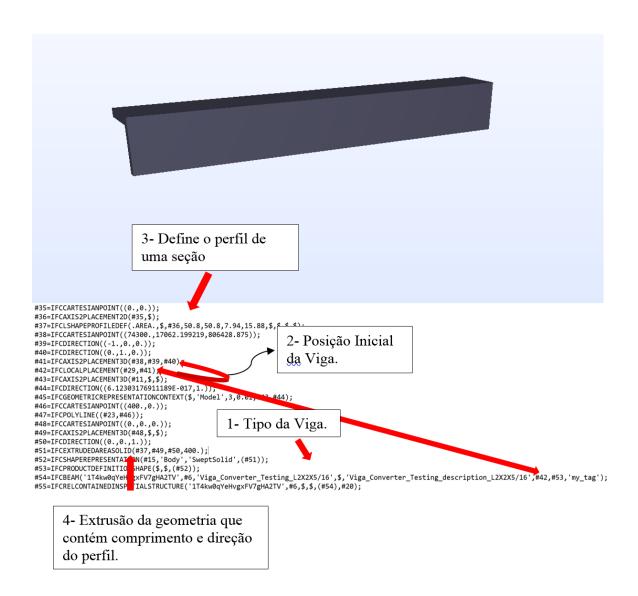


Figura 30: Viga modelada para ilustração sobre o processo de leitura de uma viga no IFC.

1. Ler o tipo da viga:

- Encontrar a classe *IfcBeam* e ler o atributo *ObjectType* do tipo *string*;
- Salvar esse atributo em uma variável do tipo *string*.

2. Ler a posição inicial da viga:

- Encontrar a classe IfcBeam e ler o atributo ObjectPlacement que vai ser o ponteiro para a classe IfcLocalPlacement;
- Usar o ponteiro para encontrar a linha onde se encontra o IfcLocalPlacement correspondente à viga;
- Acessar o segundo atributo da classe *IfcLocalPlacement* que é o ponteiro para a classe *IfcAxis2Placement3D*;

- Usar o ponteiro para localizar a classe ou entidade *IfcAxis2Placement3D* e
 ler o primeiro atributo que apontará para a classe *IfcCartesianPoint*;
- Ir na linha da classe *IfcCartesianPoint* e ler as coordenadas X, Y, Z do ponto inicial da viga.
- 3. Definir a seção do perfil preenchendo os atributos da classe *IfcLShapeProfileDefinition* (utilizada no exemplo, destacado na Figura 29).
 - o Ler profundidade da seção (*Depth*);
 - o Ler largura da seção (Width);
 - Ler espessura da seção (Thickness);
 - Ler centro de gravidade da seção em relação ao eixo x (CentreOfGravityInX).
- 4. Encontrar a profundidade e a direção do perfil para realizar a extrusão até o ponto final da viga:
 - Encontrar a classe IfcBeam e ler o atributo Representation que vai ser o ponteiro para a classe IfcProductDefinitionShape;
 - Percorrer as linhas deste padrão até encontrar a correspondente ao
 IfcProductDefinitionShape e ler o terceiro atributo (*HasShapeAspects*) que
 apontará para a classe *IfcShapeRepresentation*;
 - Ao acessar o *IfcShapeRepresentation* ler o quarto atributo que apontará para a classe *IfcExtrudedAreaSolid* responsável pela extrusão da geometria que contém a direção e a profundidade para extrusão;
 - Ler o quarto atributo da classe *IfcExtrudedAreaSolid* que é um número que indica a profundidade da viga;
 - Ler o terceiro atributo da classe *IfcExtrudedAreaSolid* que aponta para a classe *IfcDirection* e possui a direção da viga.

A Tabela 4 detalha os atributos, requisitos, funções criadas ou utilizadas da biblioteca *IfcOpenShell* e suas respectivas descrições para ler o arquivo IFC. Os atributos no IFC têm requisitos classificados como obrigatórios ou opcionais. Desta forma, apresenta-se um pacote de informações extraído de um arquivo no formato IFC 2x3. Este pacote permite a declaração das informações mínimas extraídas do IFC, para que um outro *software* além do PDMS possa ler e traduzir os dados para sua linguagem proprietária.

Tabela 4: Classes, funções e implementação da leitura do formato IFC pelo conversor.

Atributos	Requisitos	<i>IfcOpenShell</i>
GUID	Obrigatório	ifcopenshell.guid.co
		mpress(uuid.uuid1().
		hex)
<i>IFCOwnerHistory</i>	Obrigatório	ifcFile.by_type("Ifc
		OwnerHistory")[0]
Name	Opcional	'Palavra-chave com
		o nome do tipo de
		suporte definido
		pela projetista'
Description	Opcional	'Palavra-chave com
		a descrição da viga
		definida pela
		projetista'
<i>ObjectType</i>	Opcional	'beam'
ObjectPlacement	Obrigatório	ifcFile.by_type("Ifc
J	C	BuildingStorey")
Representation	Obrigatório	ifcFile.createIfcProd
•	C	uctDefinitionShape(
		None,None,[new_sh
		ape2])
Tag	Opcional	'String com Tag
	•	único definido pela
		projetista'

Nesse contexto, após apresentar as informações provenientes do IFC necessárias à análise estrutural de um perfil, na Tabela 5 são demonstradas as informações no formato Datal e sua respectiva sequência de classes relacionadas no IFC, para realizar a conversão bidirecional. Cabe ressaltar, que a carga da tubulação deve ser preenchida manualmente pelo projetista estrutural, pois o IFC não permite incorporar essa informação.

Tabela 5: Tabela de atributos para conversão bidirecional do Datal ou IFC.

Atributos	Datal	IFC
Tipo de Perfil	GTYP ou SPRE SPCOMPONENT	Tratamento da Palavra-chave do IFCBEAM.ObjectType para compatibilização com a norma.
Posição Inicial	POSS	IFCBEAM.ObjectPlacement→ IFCLOCALPLACEMENT.RelativePlacement→ IFCAXIS2PLACEMENT3D.Location→ IFCCARTESIANPOINT.Coordinates
Direção	(POSE – POSS)/ Sqrt((POSE – POSS)^2)	IFCBEAM.ObjectPlacement → IFCLOCALPLACEMENT.RelativePlacement → IFCAXIS2PLACEMENT3D.RefDirection
Comprimento	Sqrt((POSE – POSS)^2)	IFCBEAM.Representation → IFCPRODUCTDEFINITIONSHAPE.Representations → IFCSHAPEREPRESENTATION.Items → IFCEXTRUDEDAREASOLID.Depth
Material	SPRE SPCOMPONENT	IFCRELASSOCIATESMATERIAL.RelatedObject = IFCBEAM → IFCRELASSOCIATESMATERIAL.RelatingMaterial = IFCMATERIAL

A estratégia de implementação utilizou as classes para cada tipo de perfil, conforme pode-se exportar de vários *softwares* Open BIM proveniente de seus catálogos. Por exemplo, o Perfil "I" é definido pela classe *IfcIShapeProfileDef*. Essa classe é herdeira da classe *IfcParameterizedProfileDef*, ou seja, define todos os perfis mais comumente usados contidos em catálogo, isto é, são armazenadas todas as informações necessárias para a sua representação. Além da classe *IfcIShapeProfileDef*, o IFC possui diversas outras classes para representar outros tipos de perfis como, por exemplo, o perfil "L", perfil "U", perfil circular, entre outros.

Outra forma de se representar um perfil é utilizando a classe *IfcArbitraryClosedProfileDef*, ou seja, o perfil precisa ser criado a partir de polilinhas ponto a ponto para desenhar um objeto em 2D. Nessa hipótese o IFC representa o elemento em 2D, podendo realizar a extrusão linear de uma seção transversal, ou seja, o perfil em 3D, por meio da classe *IfcExtrudedAreaSolid*. Assim, as informações de altura, largura, espessura, entre outros, não são coletados do catálogo e ficam implícitos à geometria.

Para extrair essas informações nos perfis que estão definidos utilizando a classe *IfcArbitraryClosedProfileDef*, é necessário maior esforço para ler a geometria e extrair graficamente essas informações, ainda assim não seria compatível com o formato Datal que faz referência ao catálogo. Neste estudo, foi possível perceber que dependendo de como o perfil é definido, o *software* gerador do IFC pode extrair um determinado perfil sem a devida referência ao catálogo com sua chave-primária (vide Tabela 2 o atributo GUID).

Foi possível perceber essa ocorrência, por exemplo, quando um teste em um perfil do tipo "T" foi realizado. Ao invés da representação ter sido feita com a classe existente *IfcTShapeProfileDef*, que necessita das informações de altura da viga, largura da mesa, espessura da mesa e espessura da alma para ser criado, o *software* gerou uma geometria pela função *IfcArbitraryClosedProfileDef*. Dessa forma, foi necessário gerar novos pontos e criar oito novas polilinhas para fechar um polígono que define um perfil arbitrário do tipo "T". Portanto, interpretar geometrias desse tipo seria um esforço elevado e incompatível com o formato Datal, uma vez que ele necessita da compatibilização entre a norma AISC (2010) e o catálogo com sua respectiva modelagem paramétrica.

Sendo assim, após definição do detalhamento e estratégia de implementação é apresentada a interface da aplicação desenvolvida, na qual é composta de três abas, uma de conversão do *IFC* para o Datal ilustrada na Figura 31. Nesta aba pode-se apontar o caminho da pasta na rede que contenha o arquivo *IFC 2X3* a ser convertido em *Datal* (passo 1). A partir disso, basta clicar no ícone *Start* para iniciar o processo de conversão (passo 2). A leitura do IFC e escrita do Datal é detalhada no Apêndice I.

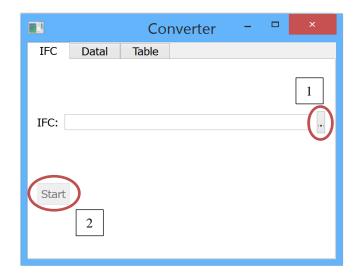


Figura 31: Interface para conversão de IFC para Datal.

Na segunda aba é realizada a conversão do formato Datal para o IFC, conforme é demonstrado na Figura 32. Deve-se apontar o caminho da pasta (passo 1) e caso necessário ou desejado pode-se apontar um *template* para criar o arquivo *IFC 2x3* com as caracteristicas de nome do responsável técnico, organização e projeto, exclusivamente definidas pelo usuário (passo 2). Após delineamento do escopo de conversão é necessário clicar em *start* para inicializar o processo (passo 3). A leitura do Datal e escrita do *IFC* é detalhada no Apêndice I.

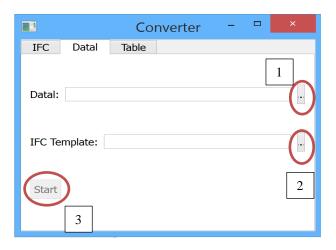


Figura 32: Interface para conversão de Datal para IFC.

A terceira aba Figura 33 é pertinente à norma de engenharia adotada pelo usuário, obrigatoriamente deve ser utilizada para verificação de conformidade dimensional dos perfis metálicos. Para isso deve-se apontar o caminho da pasta que contém a tabela da norma a ser utillizada na conversão (passo 1). A leitura de uma norma é detalhada no Apêndice I.

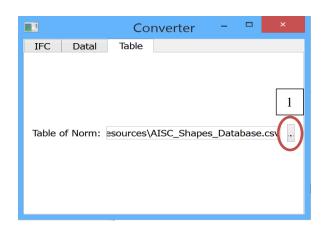


Figura 33: Interface para importação da norma utilizada na conversão.

Na seção seguinte são descritas a estratégia e detalhamento da solução desenvolvida na tradução dos dados coletados do PDMS relativos à análise de flexibilidade para o formato XML.

5.3. DESENVOLVIMENTO DE UMA TAXONOMIA E EXTRATOR DE DADOS PARA ANÁLISE DE FLEXIBILIDADE

A realização da análise de flexibilidade foi identificada no grupo focal como um dos fatores críticos de sucesso para alcançar a maturidade necessária de uma linha de tubulação, permitindo liberá-la para a fabricação. Para realizar contramedidas a este fator, optou-se pelo desenvolvimento de um extrator de dados do PDMS contendo informações necessárias à análise de flexibilidade em tubulações industriais.

Este desenvolvimento tem como objetivo propiciar uma solução computacional que explore a interoperabilidade sintática por meio de uma taxonomia padronizada em XML, extraindo e convertendo informações necessárias do PDMS para as fornecedoras dos sistemas de análise de flexibilidade, assim como, possibilitando extensões de domínio no padrão IFC. O formato IFC não foi utilizado para conversão, pois não existem classes dedicadas a tubulações industriais, por conseguinte, atributos e relacionamentos para análise de flexibilidade. Nesse contexto, é proposta na Figura 34 a arquitetura de uma solução que verifica e exporta informações necessárias à análise de flexibilidade em XML.

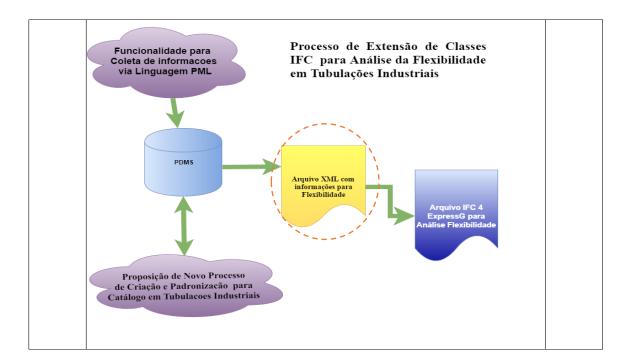


Figura 34: Arquitetura da implementação e proposição de formato IFC para análise de flexibilidade.

Mediante a arquitetura da solução apresentada na Figura 34, como premissa se destaca o preenchimento dos atributos necessários pelo projetista mecânico no catálogo do PDMS. No cenário atual essas informações são armazenadas em documentos de visualização de texto e a informação é cadastrada manualmente pelos projetistas sobre demanda de uma atividade, por exemplo, realização da análise de flexibilidade.

No cenário proposto, a linguagem PML (*Programmable Macro Language*) e a extensão XML (*eXtensible Markup Language*) são utilizadas, buscando promover mecanismos de extração de dados e um padrão sintático para inserção de tubulações industriais pelos fornecedores em ferramentas Open BIM. A implementação computacional realizada é dividida em quatro partes para coletar as informações e verificá-las:

- (1) Coleta de dados da linha de tubulação, extraindo sua nomenclatura e referência (TAG);
- (2) Informações das ramificações (*branches*), temperaturas, pressões, número do material, densidade do fluido e densidade do material de isolamento térmico;
- (3) Informações dos nós: número do nó, rigidez, peso, tipo de componente, número de referência, diâmetro externo, espessura da tubulação, sobre-espessura de corrosão, ângulo *alpha* (aplicável apenas para reduções), coordenada, raio de curvatura, tipo do raio de curvatura, espessura de isolamento térmico e classe de pressão;
- (4) Restrições de movimento nos suportes da tubulação, onde as informações são: valores das restrições de movimento (X, Y, Z), rigidez, folga e coeficiente de atrito.

Na Figura 35 são demonstradas as etapas de coleta das informações necessárias à análise de flexibilidade para sua escrita em XML. Essas informações são minimamente requeridas para tradução do PDMS para os formatos ISO 16739 (2013), ISO 15926 (2004) e qualquer outro formato no nível sintático. Vale salientar que com o extrator desenvolvido pode-se monitorar a modelagem 3D das linhas de tubulação para o correto preenchimento dos atributos definidos na taxonomia proposta, da mesma forma que é possível criar uma engenharia do conhecimento no nível semântico de interoperabilidade.

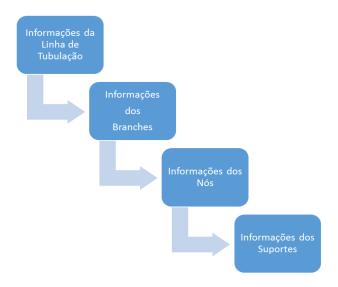


Figura 35: Hierarquia da extração de dados do PDMS via PML.

De modo a ilustrar os níveis hierárquicos de um projeto 3D de tubulações na extração dos dados para análise de flexibilidade, apresentam-se na Figura 36 os referidos pontos que necessitam de análise de tensões secundárias.

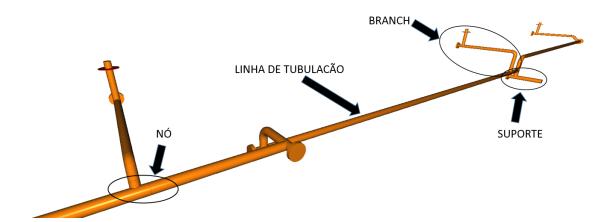


Figura 36: Diagrama de hierarquia do processo de extração e verificação de informações.

O diferencial desta solução concentra-se nas informações adicionais fornecidas se comparada ao cenário atual, em que os sistemas de análise provêm formulários para inserções manuais, por exemplo, dos atributos de pressão de teste, temperatura de operação e demais dados provenientes das especificações técnicas em formatos para visualização de texto.

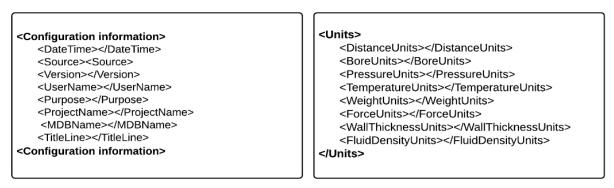
Sobretudo, a presente dissertação desenvolve um extrator de dados que auxilia a verificar o preenchimento dos atributos mínimos, propostos pela taxonomia que permite a inclusão de novas classes no IFC, conforme diversos pesquisadores realizaram para atender seus objetivos (Zhiliang *et al.*, 2011; Hegemann *et al.*, 2012; Jiao *et al.*, 2012; Janosch e Meschke, 2013; Zhang e El-Gohary, 2014, 2016; Beetz *et al.*, 2015; Motamedi *et al.*, 2016).

Contribui-se para a literatura em propor uma extensão no domínio do IFC, incluindo tubulações industriais com foco na otimização do processo da análise de flexibilidade. A contribuição prática é fornecida por meio de um extrator de dados que permite testar a completude dos atributos apresentados na taxonomia em modelos 3D do PDMS e visa traduzir os dados coletados para um padrão proposto, nesta dissertação, no formato XML. Entretanto, esse padrão exige que o projetista mecânico crie e armazene as especificações técnicas no banco de dados do catálogo de componentes do PDMS.

Essa taxonomia proposta é testada no PDMS, pois tem uma linguagem aberta que permite desenvolver um extrator e testar a exportação de seu formato gerado em XML na seção do estudo empírico III (item 5.4.3). A taxonomia busca explicitar um padrão neutro para que as fornecedoras de sistemas de engenharia e a buildingSMART possam adotar o padrão proposto. Cabe ressaltar que não existe sistemas que incorporem automaticamente o padrão e se faz necessário o desenvolvimento de tradutores em ferramentas de análise de flexibilidade no curto prazo, pois deve-se propor ao fabricante a adoção do padrão em uma nova versão do seu sistema. Desta forma, os principais agentes da transformação desse processo são os contratantes, exigindo esse padrão proposto na Figura 36 em seus contratos por meio de uma taxonomia estruturada com as informações mínimas para análise de flexibilidade.

A padronização não direciona nenhum fornecedor de *software* de análise de flexibilidade, ela propicia um padrão sintático para que as normas e fornecedores escrevam suas informações necessárias para interoperação dentro de um fluxo de trabalho. No apêndice II, pode-se verificar que o código de programação na linguagem PML coleta os dados respeitando a hierarquia com o objetivo de fornecer todas informações do banco de dados.

O fluxo de informações é bidirecional, para isso necessita-se criar e utilizar conversores para o formato proposto na Figura 37, tanto das ferramentas de modelagem quanto das ferramentas de análise de flexibilidade. No entanto, o fluxo testado e descrito no estudo empírico III, por meio de uma customização no PDMS para extração dos dados respeitando a taxonomia proposta, traduz uma linha de tubulação com dados empíricos da indústria de óleo e gás para o formato XML. As ferramentas de análise não importam esse padrão, pois trata-se de uma proposta de formato neutro para que os fornecedores adotem em suas soluções. Como ponto de atenção, as unidades de medida são definidas pelo usuário. Na Figura 37 é apresentada a taxonomia sintática de informações.



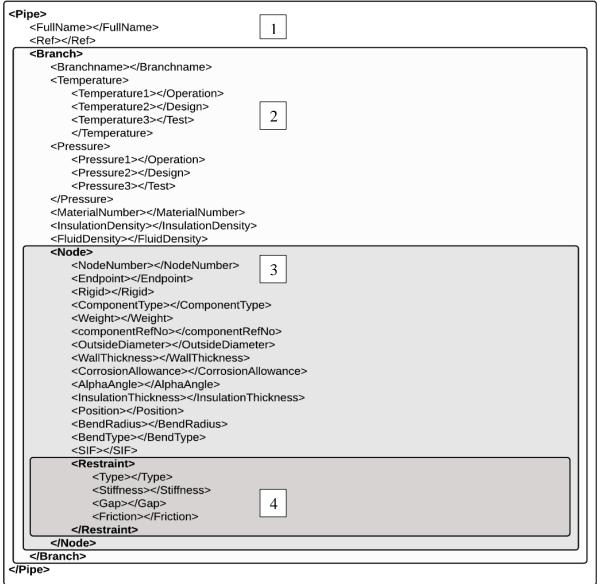


Figura 37: Taxonomia de informações para análise de flexibilidade de tubulações industriais.

O agrupamento de especificação da fonte de informação, proposta na taxonomia da Figura 37, refere-se à configuração da informação (*Configuration Information*), com objetivo de especificar e registrar os dados de saída do modelo 3D. Por exemplo: *Data Time* (data de exportação do arquivo); *Source* (nome da empresa ou instituição responsável pela informação);

Version (revisão do arquivo gerado); UserName (nome do usuário responsável pela informação); Purpose (propósito das informações); ProjectName (nome do projeto de referência); MDBName (nome do multiple database); TitleLine (título do arquivo que será exportado, a quem se destina ou para que atividade, dentre as alternativas de possíveis para casos de uso).

No agrupamento de mensuração da taxonomia proposta, especifica-se as unidades de medida, necessariamente utilizadas para análise de flexibilidade. Dentre elas: *DistanceUnits* (unidade de distância); *BoreUnits* (unidade do diâmetro nominal); *PressureUnits* (unidade de pressão); *TemperatureUnits* (unidade de temperatura); *WeightUnits* (unidade de peso); *ForceUnits* (unidade de força); *WallThicknessUnits* (unidade de espessura de parede da tubulação); e *FluidDensityUnits* (unidade da densidade do fluido).

No agrupamento seguinte, tem-se os dados de engenharia destinados à análise de flexibilidade, descritos e identificados na seguinte hierarquia: (1) pipe com FullName (nome ou "tag" da linha de tubulação) e Ref (referência do componente da linha de tubulação); (2) Branch (ramal de uma linha principal de tubulação) com *Branchname* (nome do ramal), *Temperature* 1 - Operation (temperatura de operação), Temperature2 - Design (temperatura de projeto), Temperature3 – Test (temperatura de teste), Pressure1 – Operation (pressão de operação), Pressure2 – Design (pressão de projeto), Pressure3 – Test (pressão de teste), MaterialNumber (número do material), InsulationDensity (densidade do isolamento térmico) e Fluiddensity (densidade do fluido); (3) *Node* (nó ou junção entre componentes de tubulação em que se analisa a flexibilidade de uma tubulação industrial) contendo NodeNumber (número do nó), Endpoint (ponto do nó), Rigid (rigidez), ComponentType (tipo de componente), Weight (peso), componentRefNo (número de referência), OutsideDiameter (diâmetro externo), WallThickness (espessura de parede da tubulação), CorrosionAllowance (sobre-espessura de corrosão), AphaAngle (ângulo aplicável apenas para reduções), InsulationThickness (espessura de isolamento térmico), Position (coordenadas), BendRadius (raio de curvatura), BendType (tipo do raio de curvatura) e SIF (classe de pressão); (4) Type (Valor da restrição de movimento ANC, X, Y, Z), Stiffness (restrição de dureza), Gap (folga), Friction (restrição do coeficiente de atrito). Na próxima seção, são analisados os resultados obtidos através do método de pesquisa aqui apresentado. Essa análise tem por objetivo utilizar o conversor bidirecional desenvolvido dentro do fluxo de trabalho proposto em projeto de suportes.

5.4. ESTUDOS EMPÍRICOS

A aplicação do fluxo de trabalho e tecnologia proposta para etapa 3 (projeto de suportes) é realizada no projeto 3D de uma planta industrial para o processamento de xisto. A unidade de retortagem ou pirólise faz o processamento das rochas para extração de petróleo. Trata-se de um processo de tratamento térmico, buscando a produção de óleo e gás. A aplicação na etapa 4 (análise de flexibilidade) é realizada no projeto 3D de uma unidade de destilação à vácuo. Na seção seguinte é explorada a interoperabilidade sintática, verificando a viabilidade de analisar um suporte estrutural em ferramentas Open BIM e retornar para o PDMS.

5.4.1. ESTUDO EMPÍRICO I - APLICAÇÃO DA SOLUÇÃO COMPUTACIONAL EM PROJETO DE SUPORTAÇÃO

O estudo é realizado no PDMS, selecionando linhas de tubulação e um suporte. Após a modelagem básica (etapa 1 do modelo proposto) e inserção de acessórios (etapa 2 do modelo proposto) é exportado um arquivo em formato *Datal* destas linhas de tubulação com seu respectivo suporte, conforme ilustrado em vermelho na Figura 38.

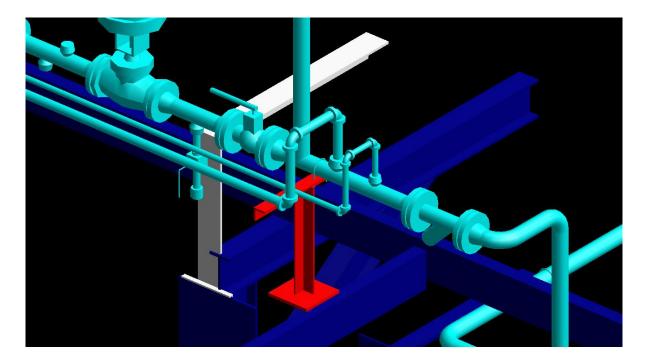


Figura 38: Projeto de tubulação com modelagem básica (etapa 1), inclusão de acessórios (etapa 2) e projeto de suportação (etapa 3) realizados.

De posse do arquivo exportado, utilizou-se o conversor desenvolvido para transformar o arquivo do formato *Datal* para *IFC 2x3*. O projeto do suporte estrutural modelado no PDMS é processado no conversor ilustrado na Figura 39, através do algoritmo (*readDatal*) apresentado no Apêndice I.

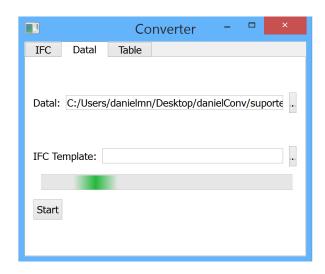


Figura 39: Processamento da conversão do *Datal* para o *IFC 2x3*.

Na transformação o algoritmo (*getAttrFromTable*) verifica a existência de compatibilidade entre os perfis metálicos dos suportes e a norma AISC (2010), gerando um arquivo *IFC 2x3*. Ao final do processo de conversão uma mensagem de conclusão (*Done*) é apresentada ao usuário, conforme ilustrado na Figura 40.



Figura 40: Resultado da conversão do *Datal* para o *IFC 2x3*.

O arquivo *IFC 2x3* gerado é importado nos softwares: *SAP 2000*, *Scia Engineering* e *Autodesk Robot*, verificando o atendimento aos requisitos mínimos (destacados na Tabela 2 da seção 5.2) para realizar análise estrutural entre os *softwares* citados. No entanto, apenas o *SAP 2000* e *Scia Engineering* obtiveram conformidade na interoperação. O *Scia Engineering* é selecionado para este experimento, em virtude de disponibilidade do acesso acadêmico. A sua leitura não gerou perdas das informações minimamente necessárias a serem trocadas para análise estrutural, conforme demonstrado na Figura 41.

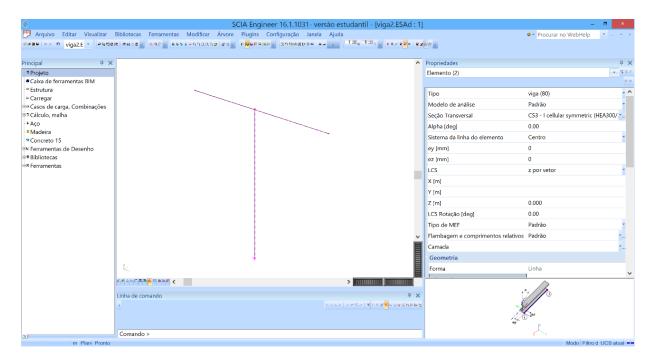


Figura 41: Modelo analítico da suportação estrutural para unidade de retortagem no *Scia Engineering*.

Vale salientar que as ligações rígidas foram mantidas, conforme demonstra a Figura 41. Desta forma o projetista estrutural deve conferir a modelagem dos pontos de ligação pelo centroide geométrico para realizar a análise estrutural. Após a leitura na ferramenta de análise estrutural, o arquivo é importado no visualizador *BIM Vision* e o resultado é exibido na Figura 42.

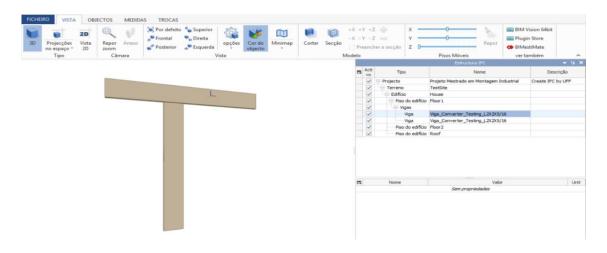


Figura 42: Visualização 3D do suporte estrutural no BIM Vision.

Pode-se perceber que a integridade dos dados do suporte estrutural foi preservada na importação do arquivo *IFC* na ferramenta de visualização *BIM Vision*. Para ilustrar as projeções no espaço tridimensional deste suporte, utilizou-se uma funcionalidade deste visualizador *Open BIM*, conforme demonstrado na Figura 43.

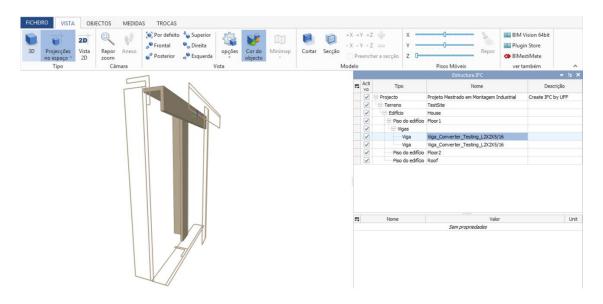


Figura 43: Visualização 3D do suporte estrutural com projeções no espaço no BIM Vision.

Após a importação nas ferramentas de análise e visualização, gerou-se um arquivo *IFC* 2x3 proveniente do *Scia Engineering*. Este arquivo é importado no conversor desenvolvido, ilustrado na Figura 44, transformando os perfis metálicos do suporte estrutural para o formato *Datal*, através dos algoritmos (*GetBeamStartPoint*; *GetBeamLenth*; *GetPipeLength*; e *GetBeamDirection*) apresentados no Apêndice I.

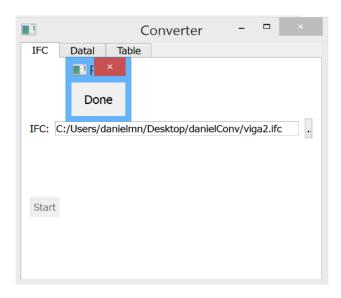


Figura 44: Processamento do suporte estrutural do Scia Engineering via IFC para o Datal.

A informação do suporte estrutural após os resultados da ferramenta de análise estrutural é retornada ao PDMS com a definição do perfil e material mais adequado, podendo ser analisada na Figura 45.

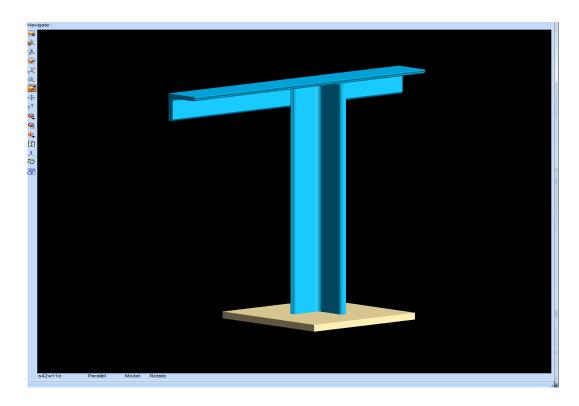


Figura 45: Importação do *Datal* com suporte estrutural aprovado no PDMS.

O suporte aprovado pela análise estrutural preservou os dados de engenharia provenientes da análise no *Scia Engineering*. Vale salientar que no cenário atual de projetos de tubulações não é realizado esse tipo de análise pelo elevado volume de suportes que necessita ser projetado para fabricação, manualmente. Esta solução proposta busca contribuir para uma maior velocidade e confiabilidade no projeto de suportes estruturais. Sobretudo, é possível realizar o detalhamento das ligações para a fabricação dos suportes na ferramenta Open BIM que for mais conveniente ao projetista estrutural.

Nesse contexto, uma desvantagem é a necessidade de inserir as cargas manualmente, uma vez que o IFC não contém classes e atributos disponíveis para tal. Cabe destacar que obrigatoriamente compartilhou-se da mesma norma disponível em ambos catálogos das soluções por *softwares* envolvidas. Em relação à carga das tubulações sobre os suportes e sua troca de informações via IFC, considera-se que este é um ponto de atenção a ser explorado pela buildingSMART para automatizar o processo. Após a realização do primeiro experimento no conversor desenvolvido, é analisado na seção seguinte um perfil metálico com maior complexidade para tradução, contendo sete suportes estruturais com características similares.

5.4.2. ESTUDO EMPÍRICO II – APLICAÇÃO DA SOLUÇÃO COMPUTACIONAL EM PROJETO DE SUPORTAÇÃO

Nessa seção são analisados suportes estruturais para tubulações e eletrocalhas (trechos para passagem de cabos elétricos), como exibido na Figura 46. Esse projeto de suportação foi realizado na mesma unidade da seção anterior de retortagem ou pirólise.

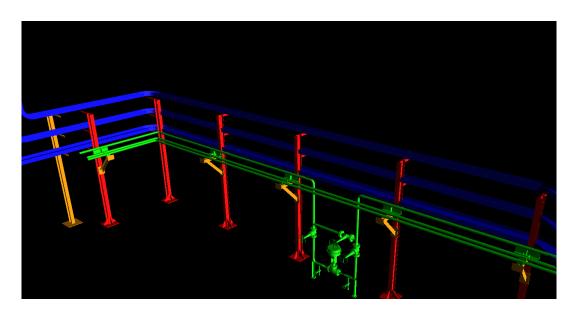


Figura 46: Visão geral dos suportes estruturais para tubulações e eletrocalhas no PDMS

Nesse contexto, delimita-se como escopo de análise um único suporte estrutural para verificar a interoperabilidade sintática e compatibilidade com padrões perante a norma AISC (2010). A partir da delimitação e exportação do *Datal*, contendo somente um suporte estrutural, este arquivo é carregado no conversor ilustrado na Figura 47.

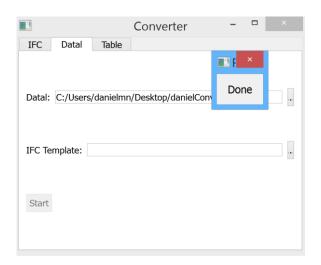


Figura 47: Processamento de arquivo *Datal* para *IFC 2x3*, contendo um suporte estrutural.

Com a conclusão do processo de conversão, o arquivo gerado (IFC 2x3) pode ser importado e visualizado no BIM Vision, possibilitando a verificação da integridade dos dados do suporte selecionado e ilustrado na Figura 48.

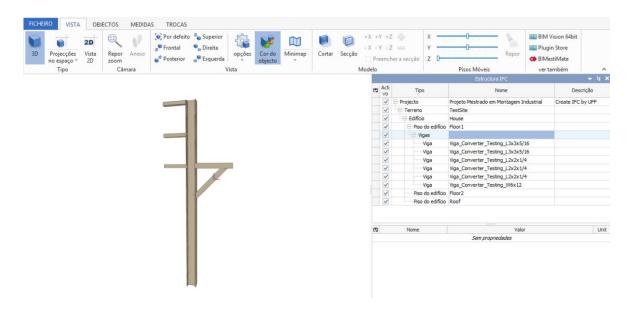


Figura 48: Suporte estrutural de tubulações e eletrocalhas no visualizador BIM Vision.

A fim de verificar se o mesmo arquivo manteria a integridade em outro visualizador Open BIM, analisando também a efetividade das ferramentas disponíveis, foi utilizado o *Solibri Model Viewer* para testar o resultado do processo de conversão. Este resultado pode ser observado na Figura 49.

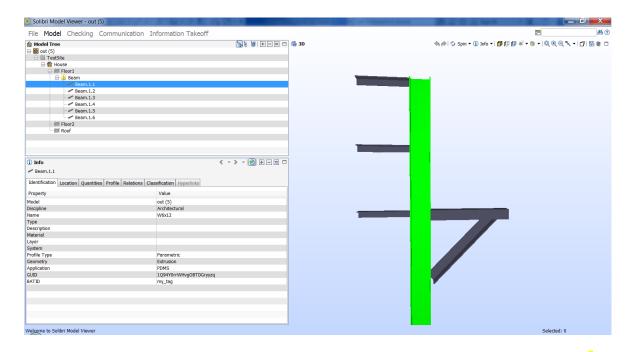


Figura 49: Resultado da importação do arquivo *IFC* gerado, no *Solibri Model Viewer*.

Após verificar a integridade da conversão do formato *Datal* para o IFC 2x3, realiza-se a importação no *SAP 2000* do mesmo arquivo IFC gerado pelo conversor para analisar se a interoperação foi realizada com sucesso. Cabe ressaltar que o objetivo é analisar se as informações foram importadas corretamente no modelo analítico, deixando disponíveis as informações no *SAP 2000* para realizar o cálculo estrutural. A Figura 50 exibe a deformação do suporte estrutural sujeito apenas à carga equivalente ao seu peso próprio. Essa análise é realizada a fim de demonstrar a viabilidade da execução do cálculo estrutural com os dados presentes no arquivo IFC gerado. Nota-se que a ligação rígida (*Rigid Link*) foi feita manualmente para possibilitar a realização da análise.

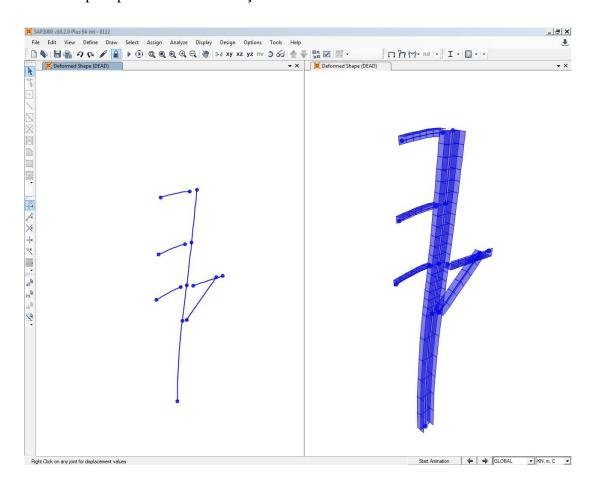


Figura 50: Resultado da importação e deformação do suporte no SAP 2000.

A partir do resultado da deformação, percebeu-se que é possível interoperar com a solução desenvolvida. No entanto, uma barreira à interoperabilidade é a consistência na importação e exportação de dados entre as diversas ferramentas testadas.

Seus tradutores são pontos de atenção para escolha das ferramentas adequadas a serem aplicadas em um fluxo de trabalho. De forma a avaliar o processo proposto em um projeto de suportação, realizou-se a importação do arquivo *IFC* gerado pelo *SAP 2000* no conversor desenvolvido para transformar o *IFC 2x3* em *Datal*, como ilustrado na Figura 51.

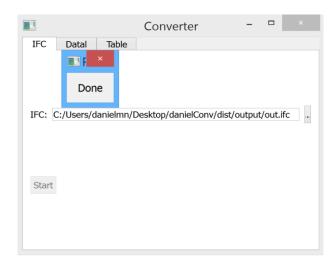


Figura 51: Processo de retorno ao PDMS a partir da conversão do IFC 2x3 em Datal.

Ao final do processo de conversão, realizou-se a importação no PDMS do arquivo *Datal* gerado. Este último passo visa retornar as informações contidas no arquivo *IFC* gerado pela ferramenta de análise estrutural. A Figura 52 apresenta o resultado da importação do referido suporte estrutural.

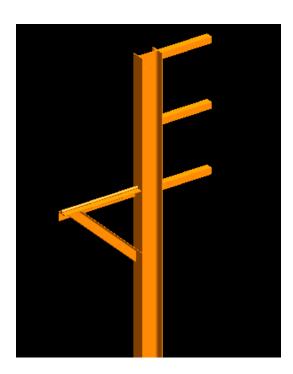


Figura 52: Retorno dos dados do suporte de tubulações e eletrocalhas para o PDMS.

A partir deste estudo, conclui-se que é possível automatizar a interface entre PDMS e ferramentas Open BIM, obter benefícios de tempo e confiabilidade das informações com mínima intervenção humana no fluxo de trabalho entre projetos de tubulação e suportes estruturais. Entretanto, deve-se ter atenção com os tradutores das ferramentas para importação e exportação seguindo o padrão imposto pelo *IFC*. Outro aspecto são os *Rigid Links*, ou seja, as ligações entre barras pelo centroide geométrico podendo ser realizada manualmente na ferramenta de análise, caso não tenha sido realizada sua ligação na ferramenta de modelagem das estruturas, conforme é realizado neste estudo empírico.

Dentro do fluxo de trabalho proposto para projeto de suportação, o conversor desenvolvido apresentou resultados satisfatórios nos estudos apresentados ao realizar a interoperação bem-sucedida de um projeto de suporte estrutural com escopo reduzido a linhas de tubulação e um suporte. Espera-se que do ponto de vista prático, especialistas possam utilizar essa metodologia e tecnologia desenvolvida, compatibilizando catálogos e normas para otimizar a interface entre tubulações e suportes em plantas industriais.

5.4.3 ESTUDO EMPÍRICO III - APLICAÇÃO DA SOLUÇÃO COMPUTACIONAL PARA ANÁLISE DE FLEXIBILIDADE

Neste estudo, é selecionado no PDMS em um projeto de uma unidade de destilação a vácuo, uma linha de tubulação com 18" de diâmetro nominal, material em aço carbono e propósito de conduzir hidrocarbonetos. De acordo com sua especificação técnica em documento 2D, a tubulação contém as seguintes características: 1,6 mm de corrosão admissível, classe de pressão 150 lbf/pol² e temperatura máxima de 400 °C, conforme ilustrado na Figura 53.

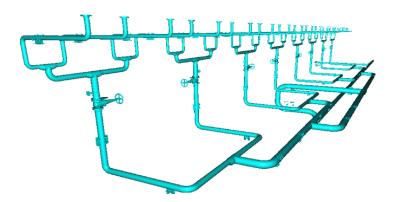


Figura 53: Linha de tubulação selecionada para extração de dados para análise de flexibilidade.

Após a etapa de seleção, utilizou-se o extrator de dados desenvolvido (conforme ilustra o código de programação no apêndice II) para verificar se os atributos mínimos de análise de flexibilidade foram preenchidos corretamente no projeto 3D em PDMS, exportando um arquivo XML. Identificou-se que poucas informações necessárias à análise de flexibilidade foram preenchidas no projeto como atributos no módulo *Design* e no catálogo do *PDMS*, principalmente: rigidez (*Stiffness*), coeficiente de atrito (*Friction*), sobre-espessura de corrosão (*CorrosionAllowance*), classe de pressão (*SIF*), número do material (*MaterialNumber*), espessura de parede (*WallThickness*), conforme apresentadas na Figura 54.

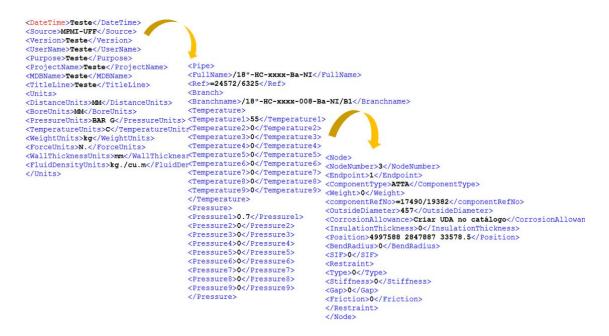


Figura 54: XML extraído de um projeto real em uma unidade de destilação à vácuo.

A análise de flexibilidade deve ser realizada em todas as condições possíveis da linha. O PDMS (projeto 3D) deve enviar à ferramenta de análise todas as temperaturas e pressões possíveis, por exemplo, de projeto, de operação, de teste, entre outras destacadas na taxonomia da seção 5.3. Sobretudo, uma solução para satisfazer as ferramentas de análise é exportar todas as pressões e temperaturas de cada ramal ou linha de tubulação proveniente do projeto 3D.

É possível utilizar o extrator de dados que realiza verificações de conformidades no preenchimento dos atributos necessários à análise de flexibilidade em projetos realizados no PDMS. Constata-se que os atributos de diversas linhas de tubulação no projeto avaliado não foram preenchidos para satisfazer automaticamente os sistemas de análise.

Entretanto, a contribuição principal encontra-se na proposição deste padrão de taxonomia, visando estimular as normas vigentes (ISO 15926 e/ou ISO 16739) e fornecedores de *software*, assim como, os contratantes a adotá-lo nos projetos de instalações industriais. Os resultados demonstraram que mudanças na cultura do projetista mecânico são necessárias, principalmente no cadastramento de informações normativas e dados de engenharia no módulo de catálogo e modelagem do *PDMS* para que o arquivo gerado pelo extrator em *XML* atenda integralmente às necessidades dos projetos de tubulação. O objetivo do desenvolvimento de um extrator de dados é comprovar a possibilidade de escrita no padrão proposto (seção 5.3), atendendo operacionalmente requisitos para o nível 2 de maturidade em interoperabilidade, o nível sintático. Vale ressaltar que o foco principal desse estudo empírico é avaliar a possibilidade de utilizar o fluxo de informações (proposto no capítulo 4) e descrever as mudanças necessárias nos hábitos de trabalho dos projetistas.

As ferramentas de análise e os padrões de interoperabilidade não estão aptos a ler ou escrever nesta taxonomia, pois trata-se de uma proposta para democratizar o intercâmbio de informações entre ferramentas e projetistas. Nessa nova abordagem, as informações podem ser traduzidas das ferramentas de modelagem paramétrica para o formato *XML* e podem ser processadas pelas ferramentas de análise estática e dinâmica da flexibilidade de tubulações. A leitura e escrita nas ferramentas de análise dependem da implementação de conversores pelos fornecedores responsáveis por cada ferramenta. Caso a ferramenta permita a criação de um conversor pelo usuário, como por exemplo o *PDMS*, programadores podem desenvolvê-lo para adequar-se ao padrão proposto.

O retorno das informações para ferramenta de modelagem paramétrica é previsto nesse padrão (apêndice II), pois se propõe um campo específico para escrever sua chave primária. O objetivo é atualizar o modelo global, no qual envolve todas disciplinas do projeto 3D. Para atualização dos resultados da análise, devem ser processados os dados geométricos e não geométricos. Portanto, como as informações exportadas para a ferramenta de análise não mudam sua chave primária, pode-se atualizar o sistema de coordenadas e dados de engenharia no projeto 3D. Nesse contexto, pretende-se explorar uma nova abordagem para realizar a análise de flexibilidade. A partir da taxonomia, pode-se propor novas classes ao padrão IFC, pois não existem pesquisas no sentido de incorporar tubulações industriais. Para tanto, deve-se solicitar a inclusão desse padrão à buildingSMART, respeitando seu procedimento e estruturação para uma criação efetiva desse domínio com seus devidos relacionamentos em versões futuras do IFC.

6. CONCLUSÃO E PESQUISAS FUTURAS

Pode-se concluir que o fluxo de trabalho proposto é um ponto de atenção para promover a mudança nos processos de trabalho com a adoção da interoperabilidade, visando otimizar recursos, diminuir retrabalho e aumentar a maturidade dos projetos de tubulação. Para isso, é imprescindível que a escolha das ferramentas de automação de projetos do empreendimento seja guiada pelas suas funcionalidades e seu atendimento pleno de conversão para os padrões adotados, porquanto, promovendo um livre mercado sem direcionar fornecedores de software e criar monopólios.

Na verificação da interoperabilidade entre as ferramentas envolvidas, observou-se como um fator crítico de sucesso a leitura e escrita do IFC. Sobretudo, preservando as características geométricas e não geométricas do catálogo de componentes que deve ser vinculada a uma norma de engenharia. Para avaliar a efetividade dos softwares testados, apresentam-se na Tabela 6 os resultados dos testes de exportação do *Autodesk Revit* e importação em diversos *softwares Open BIM*.

Tabela 6: Análise de importação e exportação de suportes modelados no *Autodesk Revit*.

Software	Tipo	Quantidade de Componentes	Quantidade de itens Exportados	Quantidade de itens Importados	% de Atributos Preenchidos
Autodesk Revit	Modelagem de Arquitetura, Estruturas e Instalações	18	18	-	-
Autodesk Robot	Análise Estrutural	-	-	3	17%
BIM Vision	Visualizador	-	-	18	100%
Solibri Model Viewer	Visualizador	-	-	18	100%
Scia Engineering	Modelagem e Análise Estrutural	-	-	18	100%
SAP 2000	Análise Estrutural	-	_	18	100%

A partir dos softwares testados (vide Tabela 6), desenvolveu-se uma solução inexistente na literatura para realizar conversão bidirecional entre ferramenta de modelagem de instalações industriais (PDMS) e ferramentas Open BIM por meio do IFC, tendo como diferencial um conversor extensível, verificando na sua tradução se os suportes estruturais estão aderentes a uma norma.

Os ganhos relativos aos projetos de suportação com a interoperabilidade sintática foram percebidos na verificação automática de qualidade dimensional dos perfis, considerando aspectos normativos, na interoperabilidade entre o PDMS e ferramentas Open BIM para verificar ou realizar o cálculo estrutural, assim como, na visualização 3D dos resultados em ferramentas de livre acesso (*open source*), portanto, evidenciando a viabilidade da etapa 3 do fluxo de maturidade proposto com seus resultados analisados nos estudos empíricos I e II. As desvantagens no processo foram constatadas na falta de classes e atributos para inclusão das cargas no IFC, fazendo com que essa definição ainda seja realizada manualmente.

Vale salientar também que poucos tradutores das diversas soluções por *softwares* envolvidas obtiveram sucesso na importação e exportação deste padrão, sendo assim a efetividade na tradução torna-se um critério de escolha das ferramentas para se utilizar o fluxo de trabalho proposto. Uma limitação deste conversor refere-se à necessidade dos sistemas envolvidos na interoperabilidade compartilharem da mesma norma de projeto, inerente a cada ferramenta de análise estrutural. Essa limitação tornou-se uma estratégia adotada para garantir a compatibilidade dos catálogos entre *softwares* e normas. Os catálogos no PDMS são referenciados pelo formato Datal e devem ter compatibilidade com a mesma norma adotada no projeto para que a implementação computacional funcione. Em trabalhos futuros poderia ser abordada a interoperabilidade entre catálogos e diferentes normas para que as ferramentas sejam cada vez mais completas, atendendo requisitos técnicos das diferentes normas existentes.

Na etapa 4 do fluxo de trabalho proposto, para realizar a análise de flexibilidade com dados provenientes do modelo 3D, deve ser modificada a cultura organizacional do preenchimento de atributos no projeto 3D. Uma vez que no estudo empírico III os documentos com especificações técnicas estavam desconexos em relação aos catálogos do modelo 3D. Os dados extraídos dos projetos reais já realizados não foram suficientes para serem consumidos pelas ferramentas de análise sem a intervenção humana, na pretendida interoperabilidade sintática.

Entretanto, uma vez que o IFC não contém classes e atributos dedicados as tubulações industriais, o objetivo tornou-se a proposição de uma padronização das informações mínimas necessárias, por meio de uma taxonomia destinada à buildingSMART para uma possível extensão do IFC em versões futuras, contemplando a área de montagem industrial em seu domínio.

As tubulações industriais têm forte correlação com estruturas, sendo diretamente dependentes para o sucesso de um projeto em plantas industriais. Esse formato é fornecido também para que pesquisas futuras possam desenvolver extratores de dados de outras ferramentas da automação de projetos com referência à chave primária de seus respectivos catálogos.

O IFC mostrou-se uma linguagem complexa com diversas classes sendo relacionadas de forma recursiva. Nota-se que essa linguagem deve ser aprimorada para evitar inconsistências tanto nas soluções por softwares certificadas quanto em novas áreas ou sistemas que pretendem obter seus benefícios. Por esse motivo a área de montagem industrial deve ter atenção com o uso do IFC em todas as disciplinas de uma planta industrial. Portanto, deve-se apenas propor extensões desse modelo no curto prazo, fundamentalmente, quando as disciplinas se relacionarem com a área de estruturas, pois o IFC tem ampla utilização na construção civil.

Um ponto de atenção no software desenvolvido se refere aos atributos necessários no IFC a cada perfil, provenientes de catálogos com modelagem paramétrica, por exemplo. Caso o usuário necessite adicionar perfis que não estão presentes nessa solução, deve-se incrementar o código de programação na linguagem Python. Os perfis contemplados nessa versão são: *Rect_Section, C_Section, L_Section e W_Section*.

Em pesquisas futuras, a norma ISO 15926 (2004) pode aproveitar as lições aprendidas desta dissertação, assim como os pacotes de informações definidos e desenvolver tradutores para o formato IFC. Cabe salientar que a ISO 15926 (2004) explora a interoperabilidade semântica um nível acima de maturidade em interoperabilidade e é destinada a plantas industriais. Entretanto, não se encontra com maturidade no curto prazo para ser utilizada pelas soluções por *softwares* disponíveis. Portanto, o uso do IFC no curto prazo para interligar as disciplinas de tubulações industriais e estruturas torna-se um viável candidato para otimizar fluxos de trabalho que envolvem a relação entre tubulações e suportes.

REFERÊNCIAS

ABANDA F.H., VIDALAKIS C., OTI A.H., TAH J.H.M., A critical analysis of Building Information Modelling systems used in construction projects, Advances in Engineering Software, V.90, 2015, p. 183-201.

AISC - AMERICAN INSTITUTE OF STEEL CONSTRUCTION. Code of Standard Practice for Steel Buildings and Bridges, 2010.

ALLISON, I. Organizational factors shaping software process improvement in small-medium sized software teams: A multi-case analysis. Quality of Information and Communications Technology (QUATIC), 2010 Seventh International Conference on the. IEEE, 2010.

AMORIM, S. R. L. e KASSEM, M.,Relatório BIM: Building Information Modeling no Brasil. Projeto Apoio aos Diálogos Setoriais União Europeia – Brasil, 2014.

ANDRADE, M. L. V. X; RUSCHEL, R. C. BIM: conceitos, cenários das pesquisas publicadas no Brasil e tendências. In: SIMPÓSIO BRASILEIRO DE QUALIDADE DE PROJETOS, São Carlos. Anais. São Carlos: RiMa Editora, v.1, 2009, p. 602-613.

ARAYICI, Y.; COATES, P.; KOSKELA, L.; KAGIOGLOU, M.; USHER, C.; O'REILLY, K. Technology adoption in the BIM implementation for lean architectural practice. Automation in Construction, v. 20, no. 2, 2011, p. 189-195.

BABIČ, N.C.; PODBREZNIK, P.; REBOLJ, D. Integrating resource production and construction using BIM. Automation in Construction, v. 19, no. 5, 2010, p. 539-543.

BEETZ, J.; COEBERGH, W.; BOTTER, R.; ZLATANOVA, S., DE LAAT, R. Interoperable data models for infrastructural artefacts - A novel IFC extension method using RDF vocabularies exemplified with quay wall structures for harbors. eWork and eBusiness in Architecture, Engineering and Construction, 2015, p. 135-140.

BRINER, R.B.; DENYER, D. Systematic Review and Evidence Synthesis as a Practice and Scholarship Tool. Handbook of Evidence-based Management: Companies, Classrooms and Research, 2012, p. 112-129.

CHEN, L.J.; LUO, H. A BIM-based construction quality management model and its applications. Automation in Construction, v. 46, 2014, p. 64–73.

CHENG J.C.P.; DAS, M. A bim-based web service framework for green building energy simulation and code checking. Journal of Information Technology in Construction, v. 19, 2014, p. 150–168.

DAVIES, R.; HARTY, C. Implementing 'site BIM': A case study of ICT innovation on a large hospital project. Automation in Construction, v. 30, 2013, p. 15-24.

EADIE, R.; BROWNE, M.; ODEYINKA, H.; MCKEOWN, C.; MCNIFF, S. BIM implementation throughout the UK construction project lifecycle: An analysis. Automation in Construction, v. 36, 2013, p. 145-151.

EASTMAN, C. M.; TEICHOLZ, P.; SACKS, R.; LISTON, K. BIM handbook: A guide to building information modeling for owners, managers, architects, engineers and contractors, 2 ed. Hoboken: John Wiley & Sons, 2011.

EASTMAN, C. M.; JEONG, Y.; SACKS, R.; KANER, I. Exchange model and exchange object concepts for implementation of national BIM standard. Journal of Computing in Civil Engineering, v. 24, 2010, p. 25–34.

FERREIRA, S. L. Da engenharia simultânea ao modelo de informações de construção (BIM): contribuição das ferramentas ao processo de projeto e vice-versa. In: Workshop Brasileiro de Gestão do Processo de Projeto na Construção de Edifícios, Curitiba, 2007.

FILHO, A. C.; SCHEER, S. Metacompilação de classes para acesso a modelos ifc e sugestões para criação de classes para acesso em alto nível. Gestão & tecnologia de projetos, v.4, 2009, p.112 - 138.

GALLAHER, M.P.; O'CONNOR, A.C.; DETTBARN, J.L.J.; GILDAY, L.T. 04-867, Cost Analysis of Inadequate Interoperability in the U.S. Capital Facilities Industry, 2004.

GRENNHALGH, T.; TAYLOR, R. Papers go beyond numbers (qualitative research). British Medical Journal, v. 315, no. 7110, 1997, p. 740-743.

GÖKÇE, K. U.; GÖKÇE, H. U.; KATRANUSCHKOV. IFC-Based Product Catalog Formalization for Software Interoperability in the Construction Management Domain. *Journal of Computing in Civil Engineering*, 27, 2013, p. 36-50.

GOVERNO DO ESTADO DE SANTA CATARINA. Caderno de Apresentação de Projetos

em BIM. Secretaria de Estado do Planejamento de Santa Catarina. 2015. p.5.

GRILO, A.; JARDIM-GONCALVES, R. Value proposition on interoperability of BIM and collaborative working environments. Automation in Construction, v. 19, no. 5, 2010, p. 522-530.

GU, N.; LONDON, K. Understanding and facilitating BIM adoption in the AEC industry. Automation in Construction, v. 19, no. 8, 2010, p. 988-999.

HEGEMANN. F., K. LEHNER.; M. KÖNIG. IFC-based product modeling for tunnel boring machines. eWork and eBusiness in Architecture, Engineering and Construction, 2012, p. 289 -296.

HU, Z.; ZHANG, J. BIM- and 4D-based integrated solution of analysis and management for conflicts and structural safety problems during construction: 2. Development and site trials. Automation in Construction, v. 20, no. 2, 2011, p. 155-166.

HM GOVERNMENT CONSTRUCTION 2025, Industrial Strategy: government and industry in partnership,2013. Disponível em: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/210099/bis-13-955-construction-2025-industrial-strategy.pdf acessado em: 10/12/2016.

IRIZARRY, J.; KARAN, E. P.; JALAEI, F. Integrating BIM and GIS to improve the visual monitoring of construction supply chain management. Automation in Construction, v. 31, 2013, p. 241-254.

ISIKDAG, U.; UNDERWOOD, J. Two design patterns for facilitating Building Information Model-based synchronous collaboration. Automation in Construction, v. 19, no. 5, 2010, p. 544-553.

ISO 15926, Industrial automation systems and integration — Integration of life-cycle data for process plants including oil and gas production facilities, 2004.

ISO 16739, Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries, 2013.

JANOSCH, S.; MESCHKE, G. Process-oriented numerical simulation of mechanized tunneling using an IFC-based tunnel product model. In: Proceedings of the 13th International Conference

on Construction Applications of Virtual Reality. London, UK. 2013.

JIAO, Y.; WANG, Y.; ZHANG, S.; LI, Y.; YANG, B.; YUAN, Lei. A cloud approach to unified lifecycle data management in architecture, engineering, construction and facilities management: Integrating BIMs and SNS. Advanced Engineering Informatics, v. 27, no. 2, 2012, p. 173-188.

KHOSROWSHAHI, F.; ARAYICI, Y. Roadmap for implementation of BIM in the UK construction industry. Engineering, Construction and Architectural Management, v. 19 no. 6, 2012, p. 610 – 635.

KIM, C.; KIM, C.; SON, H. Automated construction progress measurement using a 4D building information model and 3D data. Automation in Construction, v. 31, 2013, p. 75-82.

KLEIN, L.; LI, N.; BECERIK-GERBER, B. Imaged-based verification of as-built documentation of operational buildings. Automation in Construction, v. 21, no. 1, 2012, p. 161-171.

KUMAR, S.S.; CHENG, J.C.P. A BIM-based automated site layout planning framework for congested construction sites. Automation in Construction, v. 59, 2015, p. 24–37.

LAUD, A., Interoperability between IFC's (ISO 16739) and ISO 15926, 2013.

LEE, S. I.; BAE, J. S.; CHO, Y. S. Efficiency analysis of Set-based Design with structural building information modeling (S-BIM) on high-rise building structures. Automation in Construction, v. 23, 2012, p. 20-32.

LEITE, F.; AKCAMENTE, A.; AKINCI, B.; ATASOY, G.; ATASOY, G.; KIZILTAS, S. Analysis of modeling effort and impact of different levels of detail in building information models. Automation in Construction, v. 20, no. 5, 2011, p. 601-609.

LI, J.; KIM, I.; LEE, S.; HAN, S.; LEE, C.; CHEON, S.; LEE, W.; AN, K.; CHO, G.; HWANG, J.; MUN, D. Sharing piping CAD models of ocean plants based on international Standards. Journal of Marine Science and Technology, v. 16, no. 1, 2011, p. 76-83.

LIEBICH, T; WIX, J. Integrating use case definitions for IFC developments, in: Ework and Ebusiness in Architecture, Engineering and Construction, Proceedings of 7th European Conference on Product and Process Modelling, Sophia Antipolis, France. 2008, p. 637–645.

LIEBICH, T. IFC for INFRA structure. INFRA-BIM Workshop., Helsinki. Building Smart. International home for openBIM, 2013.

LINDEROTH, H.C.J. Understanding adoption and use of BIM as the creation of actor networks. Automation in Construction, v. 19, no. 1, 2010, p. 66-72.

MAHALINGAM, A.; KASHYAP, R.; MAHAJAN, C. An evaluation of the applicability of 4D CAD on construction projects. Automation in Construction, v. 19, no. 2, 2010, p.148-159.

MACHADO, A.B; ABREU, J.C, Aplicação de Softwares Integrados para o Desenvolvimento de Projetos de Instalações Industriais, Trabalho de Conclusão de Curso (Engenharia Química), Universidade Federal Fluminense, Niterói, 2011.

MANZONI, A. Desenvolvimento de um Sistema Computacional Orientado a Objetos para Sistemas Elétricos de Potência: Aplicação a Simulação Rápida e Análise da Estabilidade de Tensão. Tese (Doutorado em Engenharia Elétrica) - Universidade Federal do Rio de Janeiro, COPPE, Rio de Janeiro, 2005.

MILLER, P. Interoperability. What is it and why should I want it? Ariadne, n.24, 2000. Disponível em: http://www.ariadne.ac.uk/issue24/interoperability/. Acesso em: 27 de maio de 2016.

MOTAMEDI, A.; SOLTANI, M.M.; SETAYESHGAR, S.; HAMMAD, A. Extending IFC to incorporate information of RFID tags attached to building elements. Advanced Engineering Informatics, v. 30, no. 1, 2016, p. 39-53.

NEDERVEEN, G. A.V.; F. P. TOLMAN, Modelling multiple views on buildings. Automation in Construction, v. 1, no. 3, 1992, p. 215-224.

NEVES, J. L. Pesquisa qualitativa: características, usos e possibilidades. Caderno de pesquisas em administração. São Paulo, v. 1, no. 3, 1996. p. 1-4.

OLIVEIRA, F.L. O Processo da Automação de Projetos. 1995. Dissertação (Mestrado em Engenharia de Produção) - Universidade Federal do Rio de Janeiro.

OLIVEIRA, M.; FREITAS, H.M.R. Focus Group – pesquisa qualitativa: resgatando a teoria, instrumentalizando o seu planejamento. Revista de Administração, v. 33, no. 3, 1998, p. 83-91.

PAPADOPOULOS, N. A.; SOTELINO, E. D.; MARTHA, L. F.; NASCIMENTO, D. L. M.; FARIA, P. S.; Avaliação da integração entre uma plataforma BIM e uma ferramenta de análise estrutural. Sistemas & Gestão, v. 12, n. 1, 2017, p. 108-16.

PARK, C. S.; LEE, D.Y.; KWON, O.S.; WANG, X. A framework for proactive construction defect management using BIM, augmented reality and ontology-based data collection template. Automation in Construction, v. 33, 2013, p. 61-71.

PAUWELS, P. D.; DEURSEN, V.R.; ROO, V.J.D.; MEYER, R.D.; WALLE, V.J.; CAMPENHOUT, J.V. A semantic rule checking environment for building performance checking. Automation in Construction, v. 20, no. 5, 2011, p. 506-518.

PEREIRA, C. A.; CORREA, A. C. C. D.; YOGUI, R.; LIMA, C. M. Interoperability Among Engineering Systems and Their Relevance to the Effectiveness of the Engineering Projects Life Cycle Regulation, Well Succeeded Examples and Proposed Actions. 21st Brazilian Congress of Mechanical Engineering, Natal, 2011.

PEREIRA, C.A.; CLASEN, C. Interoperabilidade de Informações de Engenharia via ISO 15926, 1º Workshop de Automação de Projetos do Complexo Petroquímico do Rio de Janeiro, Itaboraí, 2013.

PILEHCHIAN, B.; STAUB-FRENCH, S.; NEPAL, M.P.A. conceptual approach to track design changes within a multi-disciplinary BIM environment. Canadian Journal of Civil Engineering, v.42, no. 2, 2015, p. 139-152.

POPOV, V.; JUOCEVICIUS, V.; MIGILINSKAS, D.; USTINOVICHIUS, L.; MIKALAUSKAS, S. The use of a virtual building design and construction model for developing an effective project concept in 5D environment. Automation in Construction, v. 19, no. 3, 2010, p. 357-367.

PORWAL, A.; HEWAGE, K. N. Building Information Modeling (BIM) partnering framework for public construction projects. Automation in Construction, v. 31, 2013, p. 204-214.

RIBEIRO, J. Grupos focados: teoria e aplicações. Porto Alegre: FEENG, 2003.

REDMOND, A.; HORE, A.; ALSHAWI, M.; WEST, R. Exploring how information exchanges

can be enhanced through Cloud BIM. Automation in Construction, v. 24, 2012, p. 175-183.

RODRIGUES, D.A. Desenvolvimento de aplicativos em Sistemas de CAD-3D para extração de dados de processos industriais, Trabalho de Conclusão de Curso (Engenharia Mecânica), Universidade Federal Fluminense, Niterói, 2008.

SACKS, R.; RADOSAVLJEVIC, M.; BARAK, R. Requirements for building information modeling based lean production management systems for construction. Automation in Construction, v. 19, n. 5, p. 641-655, 2010.

SANTOS, E.M. Desenvolvimento e Padrões de Interoperabilidade em Governo Eletrônico no Brasil. Tese (Doutorado em Administração). Programa de Pós-Graduação em Administração, USP, São Paulo, 2008.

SAYÃO L. F.; MARCONDES, C.H. O desafio da interoperabilidade e as novas perspectivas para as bibliotecas digitais. Revista TransInformação, Campinas, v. 20, no. 2, 2008, p. 133-148.

SHEN, W.; SHEN, Q.; SUN, Q. Building Information Modeling-based user activity simulation and evaluation method for improving designer-user communications. Automation in Construction, v. 21, no. 1, 2012, p. 148-160.

SILVA, E.B.A. Desenvolvimento de aplicativo em sistema CAD-3D para medição de produtividade de usuários, Trabalho de conclusão de curso (Graduação em Engenharia Civil), Universidade Federal Fluminense, Niterói, 2013.

SINGH, V.; G.U.N.; WANG, X. A theoretical framework of a BIM-based multi-disciplinary collaboration platform. Automation in Construction, v. 20, no. 2, 2011, p. 134-144.

TELLES, P.C.S. Tubulações industriais - Cálculo, 9 ed. Rio de Janeiro: LTC, 2004.

TOLK, A.; MUGUIRA, J.A.; The levels of conceptual interoperability model. Fall Simulation Interoperability Workshop. Orlando: Florida, USA: Simulation Interoperability Standards Organization, 2003.

TORTORELLA, G. L.; FOGLIATTO, F.S.; RIBEIRO, J.L.D. Identificação de fatores que afetam a sustentabilidade de melhorias em células de manufatura usando grupos focados. In: XXVIII Encontro Nacional de Engenharia de Produção. Rio de Janeiro, 2008, pp. 1-12.

VENUGOPAL, M., EASTMAN, C.M.; TEIZER, J. An ontology-based analysis of the industry foundation class schema for building information model exchanges, Advanced Engineering Informatics, v. 29, 2015, pp. 940-957.

WANG W.G.; TOLK A.; WANG W.P. The levels of conceptual interoperability model: Applying systems engineering principles to M&S.Spring Simulation Multiconference (SpringSim'09). San Diego, CA, USA. 2009.

WANG, X.; LOVE, P.E.D.; KIM, M.J.; PARK, C.; SING, C.; HOU, L. A conceptual framework for integrating building information modeling with augmented reality. Automation in Construction, v. 34, 2013, p. 37-44.

XIONG, X.; ADAN, A.; AKINCI, B.; HUBER, D. Automatic creation of semantically rich 3D building models from laser scanner data. Automation in Construction, v. 31, 2013, p. 325-337.

YIN, R.K. Case study research: design and methods. 5 ed. Los Angeles, USA: SAGE Publications, 2014.

ZHANG, J.; El-GOHARY, N.M An Automated Relationship Classification to Support Semi-Automated IFC Extension. Construction Research Congress, 2016, p. 829-838.

ZHANG, J.; EL-GOHARY, N.M. Extending building information models semi-automatically using semantic natural language processing techniques. Computing in Civil and Building Engineering – In Proceedings of the 2014 International Conference on Computing in Civil and Building Engineering, 2014, p. 2246-2253.

ZHANG, S.; TEIZER, J.;LEE, J.;EASTMAN, C.M.; VENUGOPAL, M. Building Information Modeling (BIM) and Safety: Automatic Safety Checking of Construction Models and Schedules. Automation in Construction, v. 29, 2013, p. 183-195.

ZHILIANG, M.; ZHENHUA, W.; WU, Z.; ZHE, L. Application and extension of the IFC standard in construction cost estimating for tendering in China. Automation in Construction, v. 20, no. 2, 2011, p. 196-204.

Apêndice I – Codigo de Programação na linguagem Python para Projeto de Suportes

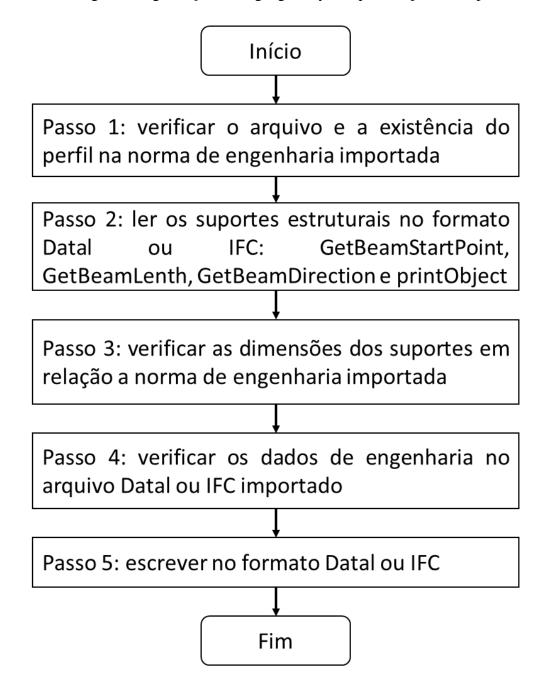


Figura 55: Diagrama do algoritmo de conversão bidirecional em datal ou IFC

```
## coding=utf-8
from bsddb.dbobj import DB
import os
import sys
import uuid
```

```
import cStringIO
import traceback
import math
import re
import numpy
import ifcopenshell
def write(filePath, inputList):
  if os.path.exists(filePath):
     os.remove(filePath)
  fileDescriptor = open(filePath, "w")
  fileDescriptor.writelines(inputList)
def read(beam, DBInput):
  ret = []
  NewObjectType = ''
  if not os.path.exists(DBInput):
     print "nao existe"
     return ret
  fileDescriptor = open(DBInput, "r")
  lines = fileDescriptor.readlines()
  for line in lines:
     pos = line.find(beam)
     if pos >=0:
        NewObjectType = line.split(';', 2)[1]
  return '/AISC/' + NewObjectType
def readDatal(inputFile):
  ret = []
  if not os.path.exists(inputFile):
     print "nao existe"
     return ret
  def getVector(strData):
     try:
        ele1 = float( re.findall(r"\d*\.*\d+", strData[strData.find(" E
") + 3: strData.find(" N ")])[0])
     except:
        ele1 = float(strData[strData.find(" E ") + 3: strData.find(" N
")])
     try:
```

```
ele2 = float( re.findall(r'' d^* ..* d^+", strData[strData.find(" N
") + 3: strData.find(" U ")])[0])
       except:
           ele2 = float(strData[strData.find(" N ") + 3: strData.find(" U
")])
       try:
           ele3 = float( re.findall(r"\d*\.*\d+", strData[strData.find(" U
") + 3:])[0])
       except:
           ele3 = float(strData[strData.find(" U ") + 3:])
       return [ele1, ele2, ele3]
   def findnth(haystack, needle, n):
       start = haystack.find(needle)
       while start >= 0 and n > 1:
           start = haystack.find(needle, start + len(needle))
       return start
   insideSCTN = False
   listSCTN = []
   for line in open(inputFile, "r").readlines():
       if line.startswith("NEW SCTN"):
           insideSCTN = True
           dictSCTN = {
               'beamName': '',
               'startPos': [],
               'endPos': [],
               'angle': 0,
           }
       elif insideSCTN:
           if line.startswith("SPRE"):
              dictSCTN['beamName'] =
line[findnth(line,'/',2)+1:].rstrip('\n')
           elif line.startswith("POSS"):
               dictSCTN['startPos'] = getVector(line)
           elif line.startswith("POSE"):
               dictSCTN['endPos'] = getVector(line)
           elif line.startswith("BANG"):
               dictSCTN['angle'] = int(re.findall(r"[+-]?\d*"
, line[5:])[0]
           elif line.startswith("END"):
               listSCTN.append(dictSCTN)
               insideSCTN = False
   return listSCTN
def getAttrFromTable(beamName, DBInput):
   ret = []
   beamAttributes = ''
   if not os.path.exists(DBInput):
       print "nao existe"
       return ret
   for line in open(DBInput, "r").readlines():
```

```
pos = line.find(';'+beamName.upper())
     if pos >=0:
       beamAttributes = line.split(';')
  return beamAttributes
def GetBeamStartPoint(beam):
  coord = beam.ObjectPlacement
  coord = coord.RelativePlacement
  coord = coord.Location
  coord = coord.Coordinates
  return coord
def GetPipeStartPoint(pipe):
  coord = pipe.Representation
  coord = coord.Representations[0]
  coord = coord.Items[0]
  coord = coord.Position
  coord = coord.Location
  coord = coord.Coordinates
  return coord
def GetBeamLenth(beam):
  beamlenth = beam.Representation
  beamlenth = beamlenth.Representations[0]
  beamlenth = beamlenth.Items
  beamlenth = beamlenth[0]
  beamlenth = beamlenth.Depth
  return beamlenth
def GetPipeLength(pipe):
  pipelenth = pipe.Representation
  pipelenth = pipelenth.Representations[0]
  pipelenth = pipelenth.Items
  pipelenth = pipelenth[0]
  pipelenth = pipelenth.Depth
  return pipelenth
def GetBeamDirection(beam):
  direction = beam.ObjectPlacement
  direction = direction.RelativePlacement
  direction = direction.RefDirection
  print direction
  if direction is None:
     direction = [1.0, 0.0, 0.0]
  else:
     direction = direction.DirectionRatios
```

```
return direction
def GetPipeDirection(pipe):
  direction = pipe.Representation
  direction = direction.Representations[0]
  direction = direction.Items[0]
  direction = direction.Position
  direction = direction.Axis
  print direction
   if direction is None:
     direction = [1.0, 0.0, 0.0]
   else:
     direction = direction.DirectionRatios
   return direction
def printObject(data):
  print "id=" + str(data.id)
  print "is ifcbeam=" + str(data.is a("ifcbeam"))
  print data.ObjectPlacement
  print data.GlobalId
  print data.Name
  print data.Description
  print data.Representation
  print data.ObjectType
def writeDatalBeam(outputFileDescriptor, startPoint, endPoint, objectType):
  outputFileDescriptor.write("NEW SCTN\n")
  outputFileDescriptor.write("GTYP BOX\n")
  outputFileDescriptor.write("DRNS E 0 N\n")
  outputFileDescriptor.write("DRNE E 0 S\n")
   #POSS E 0.0mm N 0.0mm U 0.0mm
   outputFileDescriptor.write("POSS E " + str(startPoint[0]) + "mm N " +
str(startPoint[1]) + "mm U " + str(startPoint[2]) + "mm\n")
   #POSE E 901.612121304mm N 0.0mm U 0.0mm
   outputFileDescriptor.write("POSE E " + str(endPoint[0]) + "mm N " +
str(endPoint[1]) + "mm U " + str(endPoint[2]) + "mm\n")
   outputFileDescriptor.write("JUSL NA\n")
   outputFileDescriptor.write("SPRE SPCOMPONENT "+objectType+"\n\n")
```

```
point_origin = ifcFile.by_type("IfcCartesianPoint")[0]
    beam_placement = ifcFile.by_type("IfcBuildingStorey")[0]
    ###CRIAR VIGA###
    # Beam Placement
    new cart = ifcFile.createIfcCartesianPoint(startPoint)
    new beamDirection = ifcFile.createIfcDirection(beamDirection)
    new relplac = ifcFile.createIfcAxis2Placement3D(new cart, None, Axis =
new beamDirection, RefDirection = ifcFile.createIfcDirection([0.,1.,0.]))
    objplac =
ifcFile.createIfcLocalPlacement(beam placement.ObjectPlacement,
RelativePlacement = new relplac)
    # Beam Representation part 1
    repr axis2plac3d = ifcFile.createIfcAxis2Placement3D(point origin,
None, None)
    repr new direction = ifcFile.createIfcDirection([6.12303176911189E-
17,1.])
ifcFile.createIfcGeometricRepresentationContext(None, 'Model', 3, 0.01, repr ax
is2plac3d, repr new direction)
    point origin2d = point origin =
ifcFile.by type("IfcCartesianPoint")[1]
    new cartesian_beamlength_point =
ifcFile.createIfcCartesianPoint([beamLength, 0.])
ifcFile.createIfcPolyline([point origin2d,new cartesian beamlength point])
    # Beam Representation part 2
    if str(section).find('IfcLShapeProfileDef') >=0:
        beam height = section.Depth
        beam width = section.Width
    else:
        beam height = section.OverallDepth
        beam_width = section.OverallWidth
    new axis2placement3d 2 =
ifcFile.createIfcAxis2Placement3D(ifcFile.createIfcCartesianPoint (
(0.,0.,0.)
    new extrude area = ifcFile.createIfcExtrudedAreaSolid(section,
new axis2placement3d 2, ifcFile.createIfcDirection(Z), beamLength)
    new shape2 =
ifcFile.createIfcShapeRepresentation(ifcFile.by type("IfcGeometricRepresent
ationContext")[0], 'Body', 'SweptSolid',[new_extrude area])
    new representation =
ifcFile.createIfcProductDefinitionShape (None, None, [new shape2])
    new beam = ifcFile.createIfcBeam(new guid,
owner hist, 'Viga Converter Testing '+ name,
None, 'Viga_Converter_Testing_description_'+ name,
objplac,new_representation,Tag='my_tag')
    Flr1 Container =
ifcFile.createIfcRelContainedInSpatialStructure(new guid,owner hist)
    Flr1 Container.RelatedElements=[new beam]
```

```
Flr1 Container. Relating Structure = beam placement
   # print "out " + str(os.path.join(curdir,"output\out.ifc"))
   ifcFile.write(os.path.join(curdir,"output\out.ifc"))
#################
def W Section(ifcFile, W, D, tw, tf, r):
   B1 Axis2Placement2D = ifcFile.createIfcAxis2Placement2D(
       ifcFile.createIfcCartesianPoint((0., 0.)))
   section = ifcFile.createIfcIShapeProfileDef('AREA')
   section.Position = B1 Axis2Placement2D
   section.OverallWidth = W
   section.OverallDepth = D
   section.WebThickness = tw
   section.FlangeThickness = tf
   section.FilletRadius = r
   return section
def L Section(ifcFile,W, D, t, r):
   B1 Axis2Placement2D = ifcFile.createIfcAxis2Placement2D(
       ifcFile.createIfcCartesianPoint((0., 0.)))
   section = ifcFile.createIfcLShapeProfileDef('AREA')
   section.Position = B1 Axis2Placement2D
   section.Width = W
   section.Depth = D
   section. Thickness = t
   section.FilletRadius = r
   return section
def C Section(ifcFile,W, D, tw, tf, r):
   B1 Axis2Placement2D = ifcFile.createIfcAxis2Placement2D(
       ifcFile.createIfcCartesianPoint((0., 0.)))
   section = ifcFile.createIfcUShapeProfileDef('AREA')
   section.Position = B1 Axis2Placement2D
   section.FlangeWidth = W
   section.Depth = D
   section.WebThickness = tw
   section.FlangeThickness = tf
   section.FilletRadius = r
   section. EdgeRadius = r * 0.5
   return section
def Rect Section(ifcFile,b, h):
   B1 Axis2Placement2D = ifcFile.createIfcAxis2Placement2D(
       ifcFile.createIfcCartesianPoint((0., 0.)))
   section = ifcFile.createIfcRectangleProfileDef('AREA')
   section.Position = B1 Axis2Placement2D
   section.XDim = b
   section.YDim = h
   return section
```

```
#################
def main(ifcInput, datalInput = None, DBInput =
'resources/AISC Shapes Database.csv', curdir = ""):
   try:
       if ifcInput is not None and datalInput is None:
           ifcFile = ifcopenshell.open(ifcInput)
           outputFile = open(os.path.join(curdir, "output/out.datal"),
" W " )
           print "out: " + str(outputFile)
           endpoint = [0.0, 0.0, 0.0]
           for beam in ifcFile.by type("ifcbeam"):
              print ifcFile.by type("ifcbeam")
              objectType = ';'+beam.Name+';'
              print objectType
              objectTypePDMS = read(objectType, DBInput)
               startPoint = GetBeamStartPoint(beam)
              beamLength = GetBeamLenth(beam)
              direction = GetBeamDirection(beam)
              endpoint[0] = startPoint[0] + beamLength * direction[0]
              endpoint[1] = startPoint[1] + beamLength * direction[1]
              endpoint[2] = startPoint[2] + beamLength * direction[2]
               writeDatalBeam(outputFile, startPoint, endpoint,
objectTypePDMS)
           for pipe in ifcFile.by_type("IFCFLOWSEGMENT"):
              objectType = ';'+pipe.ObjectType+';'
               startPoint = GetPipeStartPoint(pipe)
              pipelength = GetPipeLength(pipe)
              direction = GetPipeDirection(pipe)
               endpoint = [0.0, 0.0, 0.0]
               endpoint[0] = startPoint[0] + pipelength * direction[0]
               endpoint[1] = startPoint[1] + pipelength * direction[1]
               endpoint[2] = startPoint[2] + pipelength * direction[2]
               writeDatalBeam(outputFile, startPoint, endpoint,
objectType)
           outputFile.close()
       if datalInput is not None:
           #Write IFC Beam From Data
           if ifcInput is '':
               ifcInput = os.path.join(curdir, 'resources/temp.ifc')
           ifcFile = ifcopenshell.open(ifcInput)
           beamDatal = readDatal(datalInput)
           for beam in beamDatal:
               beamAttributes = getAttrFromTable(beam['beamName'],
DBInput)
               if beamAttributes != '':
                  beamSectionType = beamAttributes[0]
               else:
                  continue
               if beamSectionType == 'W':
                  # print 'W'
                  # print beamAttributes[63]
```

```
section =
W Section(ifcFile, W=float(beamAttributes[63].replace(",",".")),
D=float(beamAttributes[60].replace(",",".")),
tw=float(beamAttributes[66].replace(",",".")),
tf=float(beamAttributes[67].replace(",",".")), r=2 \star
float(beamAttributes[67].replace(",",".")))
               elif beamSectionType == 'L':
                   # print 'L'
                   # print beamAttributes[64]
                   section =
L Section(ifcFile, W=float(beamAttributes[64].replace(",",".")),
D=float(beamAttributes[60].replace(",",".")),
t=float(beamAttributes[68].replace(",",".")), r=2 *
float(beamAttributes[68].replace(",",".")))
               elif beamSectionType == 'C':
                   # print 'C'
                   # print beamAttributes[63]
                   section =
C Section(ifcFile, W=float(beamAttributes[63].replace(",",".")),
D=float(beamAttributes[61].replace(",",".")),
tw=float(beamAttributes[66].replace(",",".")),
tf=float(beamAttributes[67].replace(",",".")), r=2 *
float(beamAttributes[67].replace(",",".")))
                   section = Rect Section(ifcFile,b=200, h=400)
               #calculo do comprimento da viga dado o ponto inicial e o
final
               vetorDist = [(beam['endPos'][0]-
beam['startPos'][0]), (beam['endPos'][1]-
beam['startPos'][1]), (beam['endPos'][2]-beam['startPos'][2])]
               datalBeamLength =
\verb|math.sqrt((vetorDist[0]**2)+(vetorDist[1]**2)+(vetorDist[2]**2))|
               # calculo da direção da viga
               datalBeamDirection =
[(vetorDist[0]/datalBeamLength), (vetorDist[1]/datalBeamLength), (vetorDist[2
]/datalBeamLength)]
               createBeam(ifcFile = ifcFile, name = beam['beamName'],
section = section, startPoint = beam['startPos'], beamLength =
datalBeamLength, beamDirection = datalBeamDirection, curdir=curdir)
   except Exception, err:
       exc type, exc value, tb = sys.exc info()
       exc file = cStringIO.StringIO()
       traceback.print_exception(exc_type, exc_value, tb, file=exc file)
       exc_string = exc_file.getvalue()
       bbr = 'Excecao capturada: ' + exc string + ' ' + err.message
       print error
       return err.message
   return True
if name == " main ":
   main(None, "resources/Datal.txt")
```

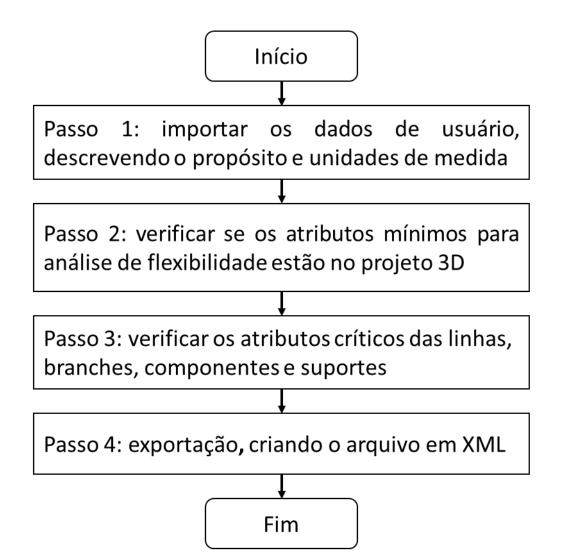


Figura 56: Diagrama do algoritmo de exportação do PDMS para o XML

```
--- Header
--- Header
--- !exportname = 'Teste'
!DateTime = 'Teste'
!Source = 'MPMI-UFF' $*
!Version = 'Teste' $* Vers棒do PDMS
!UserName = 'Teste'
!Purpose = 'Teste' $* Prop□o da an□se
```

```
!ProjectName = 'Teste'
 !MDBName = 'Teste'
 !TitleLine = 'Teste' $* Nome da an se
 !lines = array()
 !lines.append('<?xml version="1.0" encoding="utf-8"?>')
 !lines.append('<PipeStressExport ' + !exportname)</pre>
 !lines.append('<DateTime>' + !DateTime + '</DateTime>')
 !lines.append('<Source>' + !Source + '</Source>')
 !lines.append('<Version>' + !Version + '</Version>')
 !lines.append('<UserName>' + !UserName + '</UserName>')
 !lines.append('<Purpose>' + !Purpose + '</Purpose>')
 !lines.append('<ProjectName>' + !ProjectName + '</ProjectName>')
 !lines.append('<MDBName>' + !MDBName + '</MDBName>')
 !lines.append('<TitleLine>' + !TitleLine + '</TitleLine>')
 -- Units
 !DistanceUnits = 'MM'
 !BoreUnits = 'MM'
 !PressureUnits = 'BAR G'
 !TemperatureUnits = 'C'
 !WeightUnits = 'kg'
 !ForceUnits = 'N.'
 !WallThicknessUnits = 'mm'
 !FluidDensityUnits = 'kg./cu.m'
 !lines.append('<Units>')
 !lines.append('<DistanceUnits>' + !DistanceUnits + '</DistanceUnits>')
 !lines.append('<BoreUnits>' + !BoreUnits + '</BoreUnits>')
 !lines.append('<PressureUnits>' + !PressureUnits + '</PressureUnits>')
 !lines.append('<TemperatureUnits>' + !TemperatureUnits +
'</TemperatureUnits>')
 !lines.append('<WeightUnits>' + !WeightUnits + '</WeightUnits>')
 !lines.append('<ForceUnits>' + !ForceUnits + '</ForceUnits>')
 !lines.append('<WallThicknessUnits>' + !WallThicknessUnits +
'</WallThicknessUnits>')
  !lines.append('<FluidDensityUnits>' + !FluidDensityUnits +
'</FluidDensityUnits>')
 !lines.append('</Units>')
 -- Exportação
 ______
```

```
!pipes = !!collectallfor('pipe','',!!ce)
!nodenumberInt = 0
-- Dados das linhas
______
do !pipe values !pipes
 !FullName = !pipe.fullname
 !Ref = !pipe.dbref.string()
 !lines.append('<Pipe>')
 !lines.append('<FullName>' + !FullName + '</FullName>')
 !lines.append('<Ref>' + !Ref + '</Ref>')
  ______
 -- Dados dos branches
 !branches = !!collectallfor('bran','',!!ce)
 do !bran values !branches
   !lines.append('<Branch>')
    !Branchname = !bran.fullname
   !lines.append('<Branchname>' + !Branchname + '</Branchname>')
   !Temperature1 = !bran.temp.string()
   !Temperature2 = '-10000'
   !Temperature3 = '-10000'
   !Temperature4 = '-10000'
   !Temperature5 = '-10000'
   !Temperature6 = '-10000'
   !Temperature7 = '-10000'
   !Temperature8 = '-10000'
   !Temperature9 = '-10000'
   !lines.append('<Temperature>')
   !lines.append('<Temperature1>' + !Temperature1 + '</Temperature1>')
    !lines.append('<Temperature2>' + !Temperature2 + '</Temperature2>')
    !lines.append('<Temperature3>' + !Temperature3 + '</Temperature3>')
    !lines.append('<Temperature4>' + !Temperature4 + '</Temperature4>')
    !lines.append('<Temperature5>' + !Temperature5 + '</Temperature5>')
    !lines.append('<Temperature6>' + !Temperature6 + '</Temperature6>')
   !lines.append('<Temperature7>' + !Temperature7 + '</Temperature7>')
!lines.append('<Temperature8>' + !Temperature8 + '</Temperature8>')
    !lines.append('<Temperature9>' + !Temperature9 + '</Temperature9>')
   !lines.append('</Temperature>')
   !Pressure1 = !bran.pres.string()
   !Pressure2 = '0'
   !Pressure3 = '0'
   !Pressure4 = '0'
   !Pressure5 = '0'
   !Pressure6 = '0'
   !Pressure7 = '0'
   !Pressure8 = '0'
   !Pressure9 = '0'
   !lines.append('<Pressure>')
   !lines.append('<Pressure1>' + !Pressure1 + '</Pressure1>')
```

```
!lines.append('<Pressure2>' + !Pressure2 + '</Pressure2>')
      !lines.append('<Pressure3>' + !Pressure3 + '</Pressure3>')
      !lines.append('<Pressure4>' + !Pressure4 + '</Pressure4>')
      !lines.append('<Pressure5>' + !Pressure5 + '</Pressure5>')
      !lines.append('<Pressure6>' + !Pressure6 + '</Pressure6>')
      !lines.append('<Pressure7>' + !Pressure7 + '</Pressure7>')
      !lines.append('<Pressure8>' + !Pressure8 + '</Pressure8>')
      !lines.append('<Pressure9>' + !Pressure9 + '</Pressure9>')
      !lines.append('</Pressure>')
      !MaterialNumber = string(:matnum of matxt of spref of hstu of $!bran)
      !lines.append('<MaterialNumber>' + !MaterialNumber +
'</MaterialNumber>')
     handle (2,114) $* =0/0 has no attribute
      elsehandle any
      elsehandle none
        !lines.append('<InsulationDensity>' + !InsulationDensity +
'</InsulationDensity>')
      endhandle
      !FluidDensity = string(Density of sden1 of tden1 of flure of pspec of
$!bran)
      !lines.append('<FluidDensity>' + !FluidDensity + '</FluidDensity>')
      -- Dados dos componentes
      !comps = !!collectallfor('bran memb','',!!ce)
      do !component values !comps
        !lines.append('<Node>')
        !nodenumberInt = !nodenumberInt + 1
        !NodeNumber = !nodenumberInt.string()
        !lines.append('<NodeNumber>' + !NodeNumber + '</NodeNumber>')
        -- estamos exportando apenas o pa
        !Endpoint = '1'  * 0 = origin, 1 = pa, 2 = pl (for bran 1 = ph, 2 =
pt)
        !lines.append('<Endpoint>' + !Endpoint + '</Endpoint>')
        $ (
        !Rigid = 'asd'
        !lines.append('<Rigid>' + !Rigid + '</Rigid>')
        !ComponentType = !component.type
        !lines.append('<ComponentType>' + !ComponentType +
'</ComponentType>')
        !Weight = !component.brwei.string()
        handle (2,245) $* Attribute BRWEIG cannot be calculated for
element INSTRUMENT /FE-116. Attribute CMPR of SPCOMPONENT /A300/A01QBD0:100
not set or wrong for INSTRUMENT /FE-116.
        elsehandle any
        elsehandle none
          !lines.append('<Weight>' + !Weight + '</Weight>')
        endhandle
```

```
!componentRefNo = !component.ref.string()
        !lines.append('<componentRefNo>' + !componentRefNo +
'</componentRefNo>')
        !OutsideDiameter = !component.aod.string()
        handle (2,201) $* Element BRANCH /100-B-1-B2 does not have
attribute AOD
        elsehandle any
        elsehandle none
          !lines.append('<OutsideDiameter>' + !OutsideDiameter +
'</OutsideDiameter>')
        endhandle
        $ (
        !WallThickness = string(:cwall of cmpref of spref of $!component)
$* Pegar do codtub ou criar UDA no cmpref ou spref
        !lines.append('<WallThickness>' + !WallThickness +
'</WallThickness>')
        handle any
          !cn = !component.fullname
          $P $!cn
        endhandle
        $1
        !CorrosionAllowance = 'Criar UDA no cat□go'
        !lines.append('<CorrosionAllowance>' + !CorrosionAllowance +
'</CorrosionAllowance>')
        if !component.type eq 'REDU' then
          !AlphaAngle = string(atan(( abs(!component.aod - !component.lod)
/ 2 ) / (0.6 * !component.apos.distance(!component.lpos))))
          !lines.append('<AlphaAngle>' + !AlphaAngle + '</AlphaAngle>')
        endif
        !InsulationThickness = !component.inth.string()
        !lines.append('<InsulationThickness>' + !InsulationThickness +
'</InsulationThickness>')
        !Position = !component.e.string() + ' ' + !component.n.string() + '
' + !component.u.string()
        !lines.append('<Position>' + !Position + '</Position>')
        !BendRadius = string(tan(!component.adir.angle(!component.ldir) /
2 ) * !component.apos.distance(!component.pos))
        handle (2,201) $* Element TEE 1 of BRANCH /250-B-50/B1 does not
have attribute LDIR
        elsehandle none
          !lines.append('<BendRadius>' + !BendRadius + '</BendRadius>')
        endhandle
        $ (
        !BendType = !component
        !lines.append('<BendType>' + !BendType + '</BendType>')
        $)
        !SIF = string(rating of cate of catref of $!component)
        !lines.append('<SIF>' + !SIF + '</SIF>')
```

```
-- Dados dos suportes
       ______
      if !component.type eq 'ATTA' then
        --se for suporte, ou seja, se tiver geometria
        !isSupport = true
        if !isSupport then
          !lines.append('<Restraint>')
          !Type = !component.:NodeType.string()
          !lines.append('<Type>' + !Type + '</Type>')
          !Stiffness = !component.:Nodestiff.string()
          !lines.append('<Stiffness>' + !Stiffness + '</Stiffness>')
          !Gap = !component.:Nodegap.string()
          !lines.append('<Gap>' + !Gap + '</Gap>')
          !Friction = !component.:NodeFriction.string()
          !lines.append('<Friction>' + !Friction + '</Friction>')
          !lines.append('</Restraint>')
        endif
      endif
      !lines.append('</Node>')
     enddo
     !lines.append('</Branch>')
   enddo
   !lines.append('</Pipe>')
 enddo
 !lines.append('</Export>')
 ______
 -- Fim da exportação, criando o arquivo
 ______
 !file = object file('c:\aveva\pipingflexibilitypdms2xml.xml')
 !file.writefile('over',!lines)
 !filename = !file.fullname()
 !!alert.message('Arquivo
$!filename
Criado')
```