U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

O MODELO IFC COMO AGENTE DE INTEROPERABILIDADE

Aplicação ao domínio das estruturas

SÉRGIO MIGUEL FERREIRA DE PINHO

Dissertação submetida para satisfação parcial dos requisitos do grau de

MESTRE EM ENGENHARIA CIVIL — ESPECIALIZAÇÃO EM CONSTRUÇÕES

Orientador: Professor Doutor João Pedro da Silva Poças Martins

Co-Orientador: Professor Doutor João Filipe Meneses Espinheira Rio

Co-Orientador: Professor Doutor Miguel Ângelo Carvalho Ferraz

JUNHO DE 2013

MESTRADO INTEGRADO EM ENGENHARIA CIVIL 2012/2013

DEPARTAMENTO DE ENGENHARIA CIVIL

Tel. +351-22-508 1901

Fax +351-22-508 1446

✉ miec@fe.up.pt

Editado por

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Rua Dr. Roberto Frias

4200-465 PORTO

Portugal

Tel. +351-22-508 1400

Fax +351-22-508 1440

✉ feup@fe.up.pt

🌐 http://www.fe.up.pt

Reproduções parciais deste documento serão autorizadas na condição que seja mencionado o Autor e feita referência a *Mestrado Integrado em Engenharia Civil - 2012/2013 - Departamento de Engenharia Civil, Faculdade de Engenharia da Universidade do Porto, Porto, Portugal, 2013*.

As opiniões e informações incluídas neste documento representam unicamente o ponto de vista do respetivo Autor, não podendo o Editor aceitar qualquer responsabilidade legal ou outra em relação a erros ou omissões que possam existir.

Este documento foi produzido a partir de versão eletrónica fornecida pelo respetivo Autor.

Aos meus Pais e Irmão,

À minha namorada.

"Nobody said it was easy, no one ever said it would be this hard."

- Chris Martin, "The Scientist", 2002

AGRADECIMENTOS

Para a realização da presente dissertação contei com o apoio directo e indirecto de inúmeras pessoas às quais devo muito. Quero deixar expressos os meus agradecimentos, ainda que correndo o risco de injustamente não mencionar alguém, quero deixar claro que recordo sem excepção todos aqueles que contribuíram para que eu pudesse dar mais um passo nesta caminhada.

Agradeço ao meu orientador Doutor Professor João Pedro Poças Martins pela orientação prestada mas sobretudo pelo entusiasmo e interesse contagiante pelas matérias estudadas e pela liberdade e confiança que me foi depositada na realização deste trabalho. A sua atitude exemplar, domínio de conhecimentos e profissionalismo unidos à sua boa disposição são para mim um modelo a seguir.

Ao meu co-orientador Doutor Professor João Filipe Meneses Espinheira Rio, pela sua disponibilidade constante, pelo seu incentivo e igualmente pelo seu apoio e ajuda na elaboração deste trabalho. A sua indiscutível amizade, boa disposição e compreensão em momentos de consternação permitiram sempre fazer ressurgir em mim toda a motivação e autoconfiança necessária para ir um pouco mais além.

Agradeço também ao Doutor Professor Miguel Ângelo Carvalho Ferraz pelo acompanhamento prestado, pelas demonstrações mais técnicas, e pela disponibilidade e ajuda na construção do rumo que a dissertação tomou. As inúmeras conversas permitiram-me limar arestas e perceber quais as melhores direcções a seguir.

Quero aproveitar a ocasião para agradecer a toda a minha família pelo apoio incondicional, em particular aos meus pais, Sérgio Pinho e Jacinta Pinho, e irmão, Adérito Pinho, não só pelo apoio que sempre me deram ao longo deste curso de Engenharia Civil, mas também por todo o carinho e afecto que estiveram sempre presente durante toda a minha vida. Os seus testemunhos de vida foram para mim verdadeira fonte de motivação, aos quais respondi na medida do possível, com esforço e dedicação.

À minha namorada, Ana Catarina Azevedo, por ter caminhado a meu lado durante todo este percurso universitário, pela sua paciência e compreensão durante a elaboração da presente dissertação, a quem peço desculpa pelos dias sacrificados em prol da realização deste trabalho. Agradeço o seu amor e amizade e as suas palavras de motivação e confiança.

Deixo também uma palavra de apreço aos meus amigos de infância com os quais passei momentos únicos, sou obrigado a nomear os meus melhores amigos, José Nunes (Kiki) e Ricardo Pinheiro (Costa), para quem envio um agradecimento em especial.

Enfim, quero demonstrar o meu agradecimento, a todos aqueles que, de um modo ou de outro me acompanharam durante a fase do curso e tornaram possível a realização deste trabalho.

A todos o meu sincero e profundo muito obrigado.

RESUMO

A filosofia envolta no conceito BIM (*Building Information Modeling*) possibilita a junção de diferentes âmbitos do projecto num único modelo virtual, o que traz uma crescente necessidade de constante troca de informação entre os intervenientes nos projectos de edifícios. Maiores níveis de interoperabilidade significam maiores níveis de envolvimento entre os participantes na elaboração de um projecto de engenharia civil e por isso permite um maior acompanhamento do fluxo de trabalho, maior automatização de processos e maior controlo de erros e falhas imputáveis não só aquando da realização do projecto mas também ao longo do ciclo de vida da obra. Assim, a aplicação de metodologias BIM permite um aumento significativo da velocidade dos processos de projecto e construção.

O *standard* IFC enquanto agente de interoperabilidade é já implementado há mais de uma década e meia. Presentemente é exponênciada pelo surgimento do novo esquema IFC4 suportado por normalização ISO, o que representa um grande avanço da especificação IFC.

Atendendo a isto, o objectivo deste estudo passa por começar a investigar entidades IFC que permitam a definição estrutural de elementos pela sua definição analítica, função que é agora suportada pelo IFC4. Com a grande celeridade dos avanços tecnológicos torna-se necessário estar sempre um passo à frente em tudo o que diz respeito a sistemas informáticos. Portanto, no domínio IFC, a presente dissertação incidiu especialmente sobre elementos simples de estruturas de betão (pilares, pórticos, estruturas porticadas) cuja multiplicação permite a composição de estruturas de edifícios regulares. Para além da identificação das classes necessárias para a definição dos elementos estruturais que compõem uma edificação regular, fez-se também um estudo dos níveis de interoperabilidade actuais para a transferência desses dados e demonstra-se uma possibilidade de aplicação futura do modelo IFC. Essa aplicação prática das classes IFC demonstra o grande potencial da especificação, na troca de informações entre os programas de modelação e os programas de cálculo e análise estrutural. Para além de constituir um agente de interoperabilidade, pode também ser encarado como uma ferramenta BIM deste ponto de vista. Contudo tem sido alvo de criticas por parte de diversos autores no que toca à sua aplicabilidade no domínio das estruturas.

PALAVRAS-CHAVE: BIM, IFC, Interoperabilidade, Gestão de informação, Estruturas.

ABSTRACT

The philosophy around the BIM (Building Information Modeling) concept enables the joining of different project scopes in a single virtual model, which brings a growing need for constant information exchange between stakeholders in construction projects. A higher level of interoperability means higher levels of engagement among participants in the building project and therefore allows greater monitoring workflow, greater automation of processes and greater control of errors and failures throughout the life-cycle of the building. Thus, the implementation of BIM enables a significant increase of speed in project and building processes.

The IFC standard as an interoperability agent has been now implemented for over one and a half. This additional pressure now comes with the realization of the new IFC4 scheme as an ISO standardization, which represents a major breakthrough of the IFC specification.

Hereby, the aim of this study subject begin by investigate IFC entities that allow the structural definition of building elements defining their analytical functions, which is now supported by IFC4. The fast technological advances require us to be always one step ahead in all that concerns informatical sistems. Therefore, in the IFC field, this thesis focused especially on simple concrete structural elements (like footings, columns, beams and slabs) whose multiplication allows the composition of regular building structures. Beyond identifying the classes needed to define the structural elements that make up a regular building, was also made a study of the current levels of interoperability for the transfer of such data and it is also shown a possibility of future application of the IFC model. This practical application of IFC classes demonstrates the great potential within the IFC specification, in what concerns the exchange of information between modeling programs and programs for calculation and structural analysis, in addition to being an agent for interoperability can also be seen as a BIM tool by this point of view, however, has been the target of criticism by various authors in regard to its applicability in the field of structures

KEYWORDS: BIM, IFC, Interoperability, Information management, Structures.

ÍNDICE GERAL

AGRADECIMENTOS i
RESUMO iii
ABSTRACT v
ÍNDICE DE FIGURAS xi
ÍNDICE DE QUADROS xv
TERMOS E DEFINIÇÕES xvii
ABREVIATURAS E SIMBOLOS xix

1 INTRODUÇÃO 1
1.1. CONSIDERAÇÕES INICIAIS 1
1.2. ÂMBITO E OBJECTIVOS 2
1.3. ORGANIZAÇÃO DA DISSERTAÇÃO 3

2 ESTADO DA ARTE 5
2.1. GESTÃO DA INFORMAÇÃO 5
2.1.1. ASPECTOS GERAIS 5
2.1.2. DOCUMENTAÇÃO DE PROJECTO 5
2.1.3. GESTÃO DE PROJECTO (PROJECT MANAGEMENT) 6
2.2. BIM – BUILDING INFORMATION MODELING 8
2.2.1. CONCEITO 8
2.2.2. VANTAGENS E DESVANTAGENS 11
2.2.3. FERRAMENTAS BIM 13
2.3. INTEROPERABILIDADE 14
2.4. IFC – INDUSTRY FOUNDATION CLASSES 17
2.4.1. CONTEXTO HISTÓRICO 17
2.4.2. CONCEITO 17
2.4.3. NORMAS 18
2.4.4. ESTRUTURA DO IFC 20
2.4.5. O IFC NA INDÚSTRIA AEC 24

3 ESTUDO EMPIRICO DE INTEROPERABILIDADE 25

3.1. INTRODUÇÃO 25

3.2. APLICAÇÃO À REALIDADE DE 2013 25

3.3. DEFINIÇÃO DA ESTRATÉGIA DE AVALIAÇÃO 26

3.4. FUNCIONAMENTO INTERNO DOS SOFTWARES NO DOMÍNIO DA INTEROPERABILIDADE 28

3.4.1. REVIT 30

3.4.2. ROBOT 34

3.4.3. ARCHICAD 35

3.4.4. TRICALC 36

3.4.5. NIST IFC FILE ANALYSER 36

3.5. TESTES DE INTEROPERABILIDADE 37

3.5.1. CASO A – REVIT ARCHITECTURE PARA ROBOT 37

3.5.2. CASO B – REVIT STRUCTURES PARA ROBOT 39

3.5.3. CASO C – ROBOT PARA REVIT ARCHITECTURE 41

3.5.4. CASO D – ROBOT PARA REVIT STRUCTURES 42

3.5.5. CASO E – REVIT STRUCTURES PARA REVIT STRUCTURES 42

3.5.6. CASO F – ARCHICAD PARA TRICALC 43

3.5.7. CASO G – TRICALC PARA ARCHICAD/REVIT/ROBOT 44

3.5.8. CASO H – TRICALC PARA TRICALC 44

3.5.9. CASO I – IFC4 PARA REVIT/ROBOT/ARCHICAD/TRICALC 44

3.6. QUADRO RESUMO 45

4 MODELAÇÃO EM IFC (CASO DE ESTUDO) 47

4.1. DESCRIÇÃO SEMÂNTICA 47

4.1.1. OBJECTIVOS 47

4.1.2. PERSPECTIVAS FUTURAS 48

4.1.3. IMPLEMENTAÇÃO DO IFC4 48

4.2. CASO DE APLICAÇÃO 50

4.2.1. PROPOSTA 50

4.2.2. ELABORAÇÃO DO MODELO IFC 51

4.3. ESTRUTURA INTERNA DE FICHEIROS IFC 53

4.3.1. CONCEITOS BASE 53

4.3.2. SIMBOLOGIA USADA 57

4.3.3. ANÁLISE E EXPOSIÇÃO DA CRIAÇÃO DE UM FICHEIRO IFC 58
4.4. CONSIDERAÇÕES COMPLEMENTARES 78

5 PROPOSTA DE IMPLEMENTAÇÃO DE CLASSES IFC NA CRIAÇÃO DE SOFTWARES 79
5.1. ENUNCIADO DO PROBLEMA E OBJECTIVOS 79
5.2. ESTRATÉGIA DE ABORDAGEM 80
5.2.1. DETALHAR 81
5.2.2. AGRUPAR 82
5.2.3. DUPLICAR 83
5.3. PROPOSTAS DE IMPLEMENTAÇÃO 84
5.3.1. EXPANSÃO DA APLICAÇÃO "EVOLUTION BIM PAVILION" 84
5.3.2. DESENVOLVIMENTO DE APLICAÇÃO PARA A RESTITUIÇÃO DE DADOS AO FICHEIRO IFC 86
5.4. DESENVOLVIMENTOS FUTUROS 89

6 CONCLUSÕES 91
6.1. DISCUSSÃO 91
6.2. PRINCIPAIS DIFICULDADES 92
6.3. CONSIDERAÇÕES FINAIS 93

REFERÊNCIAS 97

ÍNDICE DE FIGURAS

Figura 1.1 – Modelo BIM com a inclusão de várias disciplinas. Adoptado de (Autodesk 2013). 2

Figura 2.1 – Os âmbitos dos BIM e suas inter-relações. Adoptado de SCIA Scientific Software. 6

Figura 2.2 – O BIM no ciclo de vida de um edifício. (Autodesk 2013) 9

Figura 2.3 – Exemplo de modelação de estrutura 3D a partir uma base CAD 10

Figura 2.4 – Extracção de informação diversa do modelo 3D (Graphisoft 2013) 12

Figura 2.5 – Correspondência entre o modelo BIM e a situação real (Lillja 2012). 13

Figura 2.6 – Fragmentação do processo construtivo e suas consequências. (Hipólito Sousa 2006). . 15

Figura 2.7 – Cronologia do padrão IFC. (Liebich 2013) 17

Figura 2.8 – Diagrama com as normas implementadas pela BuildingSMART. (BuildingSMART 2013). 19

Figura 2.9 – Funcionamento do conceito de camadas na arquitectura do IFC. Adoptado de (IAI 1999). 21

Figura 2.10 – Estrutura da base de dados da especificação IFC4. Adaptado de (BuildingSMART 2013). 23

Figura 2.11 – O formato IFC enquanto agente activo de interoperabilidade no openBIM 24

Figura 3.1 – O formato IFC como método genérico para transferia de dados entre dois aplicativos BIM. 27

Figura 3.2 – Diagrama de procedimento para realização de testes de interoperabilidade 28

Figura 3.3 – Ficheiro IFC exportado pelo Revit e visualizado com o Solibri Model Viewer. 29

Figura 3.4 – Ficheiro IFC exportado pelo Revit e importado pelo Archicad. 29

Figura 3.5 – Definições das condições de fronteira antes da introdução das famílias respectivas. 30

Figura 3.6 – Definições das condições de fronteira depois da introdução das famílias respectivas.... 30

Figura 3.7 – Janela do Revit para a integração de modelos do Revit no Robot e vice-versa. 31

Figura 3.8 – Opções de exportação de modelos do Revit para o Robot de forma directa. 32

Figura 3.9 – Tabela de correspondência entre as categorias do Revit e as classes IFC sustentada pelo ficheiro exportlayers-ifc-IAI.txt. 33

Figura 3.10 – Tabela de correspondência entre as classes IFC e as categorias do Revit e sustentada pelo ficheiro exportlayers-dwg-ISO13567.txt. 34

Figura 3.11 – Opções de importação de modelos do Robot para o Revit de forma directa. 35

Figura 3.12 – Lista de classes IFC exportadas pelo Archicad 35

Figura 3.13 – Exemplo de *PropertySets* aplicáveis no Archicad. 36

Figura 3.14 – Janela de aviso apresentada pelo Revit Architecture. 38

Figura 3.15 – Desconexão entre apoio, pilar e laje. 38

Figura 3.16 – Janela de verificação apresentada pelo Revit Architecture. 40

Figura 3.17 – Ligação entre elementos. 40

Figura 3.18 – Exemplo de pilares unidos estruturalmente ao eixo analítico da viga através de *Rigid Links* (representados a verde). (Autodesk 2013) 41

Figura 3.19 – Opções de tradução do modelo IFC usando o Archicad. 43

Figura 3.20 – Ferramenta de ajuste de nós dos modelos importados para o Tricalc via IFC. 44

Figura 4.1 – Sequência de palavras-chave relacionadas com o IFC 49

Figura 4.2 – Dimensões de implantação da estrutura porticada. 50

Figura 4.3 – Visualização do modelo IFC analítico através do Constructivity Model Viewer 52

Figura 4.4 – Visualização do modelo IFC analítico através do Solibri Model Viewer 52

Figura 4.5 – Estrutura tipo de um ficheiro STEP (Nour 2008) 53

Figura 4.6 – Organização de uma entidade. Adaptado (Nour 2008) 55

Figura 4.7 – Exemplo de uma entidade. 55

Figura 4.8 – Codificação base de 64 caracteres para a formulação de um IFC-GUID. (BuildingSMART 2013) 56

Figura 4.9 – Correspondência entre o GUID no ficheiro IFC e no visualizador Solibri 56

Figura 4.10 – Modelo IFC - Secção HEADER do modelo IFC criado. 59

Figura 4.11 – Aplicação - Secção HEADER do modelo IFC criado. 59

Figura 4.12 – Modelo IFC - Dados do usuário, da organização e da aplicação 59

Figura 4.13 – Aplicação - Dados do usuário, da organização e da aplicação 60

Figura 4.14 – Modelo IFC – Pontos cartesianos e direcções. 61

Figura 4.15 – Aplicação - Pontos cartesianos e direcções. 61

Figura 4.16 – Modelo IFC - Unidades 62

Figura 4.17 – Aplicação - Unidades 62

Figura 4.18 – Exemplo – Conversão de unidades. 63

Figura 4.19 – Exemplo – Unidade derivada 63

Figura 4.20 – Modelo IFC – O projecto 64

Figura 4.21 – Aplicação – O projecto 64

Figura 4.22 – Modelo IFC – O projecto 64

Figura 4.23 – Aplicação – O projecto 65

Figura 4.24 – Composição do Edifício (BuildingSMART 2013) 66

Figura 4.25 – Modelo IFC – Representação Geométrica 67

Figura 4.26 – Aplicação – Representação Geométrica 67

Figura 4.27 – Modelo IFC – Nós de ligação do Pilar 1 68

Figura 4.28 – Aplicação – Nós de ligação do Pilar 1 68

Figura 4.29 – Modelo IFC – Membro estrutural do Pilar 1 69
Figura 4.30 – Modelo IFC – Membro estrutural da Viga 1 69
Figura 4.31 – Aplicação – Membro estrutural do Pilar 1 e/ou Viga 1 69
Figura 4.32 – Modelo IFC – Membro estrutural da Laje 1 70
Figura 4.33 – Aplicação – Membro estrutural da Laje 1 70
Figura 4.34 – Modelo IFC – Atribuição dos nós e membros estruturais ao modelo de análise estrutural 70
Figura 4.35 – Aplicação – Atribuição dos nós e membros estruturais ao modelo de análise estrutural 71
Figura 4.36 – Modelo IFC – Definição da arquitectura dos pilares e algumas definições 72
Figura 4.37 – Aplicação – Definição da arquitectura dos pilares e algumas definições 72
Figura 4.38 – Relação para a contenção espacial de elementos (BuildingSMART 2013) 73
Figura 4.39 – Modelo IFC – Propriedades atribuídas ao Pilar 1 74
Figura 4.40 – Aplicação – Propriedades atribuídas ao Pilar 1 75
Figura 4.41 – Modelo IFC – Modelo de Análise Estrutural 75
Figura 4.42 – Aplicação – Modelo de Análise Estrutural 75
Figura 4.43 – Modelo IFC – Cargas estruturais 77
Figura 4.44 – Aplicação – Cargas estruturais 77
Figura 5.1 – Processo para a definição de elementos compostos. 80
Figura 5.2 – Decomposição de um pilar em classes IFC. 81
Figura 5.3 – Relações de atribuição inversa a um pilar. Adaptado de (BuildingSMART 2013). 83
Figura 5.4 – Exemplo da partilha de uma secção por duas sapatas. 83
Figura 5.5 – Aplicação *Evolution BIM Pavilion*. 84
Figura 5.6 – Pavilhão originado pela aplicação *Evolution BIM Pavilion.* 85
Figura 5.7 – Aplicação *Evolution BIM Buildings*. 84
Figura 5.8 – Estrutura originada pela aplicação *Evolution BIM Buildings.* 85
Figura 5.9 – Processo de transferência de informação entre duas aplicações usando o modelo IFC e uma aplicação de suporte. 87
Figura 5.10 – Fluxograma de acréscimo de valor ao ficheiro IFC exportado pela adição de informações. 88
Figura 5.11 – Relação entre três vertentes BIM (BuildingSMART 2013) 89
Figura 6.1 – Diagrama de maturidade BIM (Bew and Richards 2008). 93

ÍNDICE DE QUADROS

Quadro 1 – Interoperabilidade entre diferentes *softwares* 46

TERMOS E DEFINIÇÕES

Atributo – Unidade de informação dentro da entidade, definido por um tipo particular ou referenciado a uma entidade particular. Existem três espécies de atributos, os directos, os inversos e os derivados.

Atributo directo – Variável indexada na secção das definições de atributos de uma entidade. Semelhante ao termo "campo" em linguagem de programação comum.

Atributo inverso – Unidade de informação na secção de atributos de uma entidade que se define fazendo o relacionamento com uma entidade externa. Essa relação é feita colocando o identificador único dessa entidade externa (#nnnnnnnnn) num campo de atributo apropriado da entidade em causa.

Atributo derivado – Unidade de informação calculada a partir de outros atributos, usando expressões definidas no esquema. São definidos fora do ficheiro IFC e representados por um asterisco (*).

Classe Abstracta Supertipo (*Abstract Supertype*) – Classe abstracta que regula um subconjunto de entidades que lhe estão indexadas, cujas definições são vocacionadas para uma determinada área.

Classe Supertipo (*Supertype*) – Classe que regula um subconjunto de entidades que lhe estão indexadas, cujas definições são vocacionadas para uma determinada área.

Classe Subtipo (*Subtype*) – Classe que faz parte de um subconjunto de entidades, administrada por uma classe supertipo.

Controlo – Aplica os conceitos de verificação, revisão e restrição. Pode ser visto como um guia que de implementação que pretende regulamentar e especificar uma exigência que deve ser aplicada a um objecto.

Entidade (ou classe) – Classe de informação definida por atributos e restrições.

Grupo – Colecção arbitrária de objectos.

Instância – Ocorrência de uma entidade.

Intervenientes – Agentes humanos que estão envolvidos em parte ou na totalidade de um ciclo de vida de um projecto.

Processo – Acção que ocorre num projecto e que tem o intuito de adquirir, construir ou manter objectos. Um processo enquadra-se numa sequência temporária.

Produto – É um objecto físico, fabricado, fornecido ou criado para incorporar o projecto. Pode ser definido através de uma representação geométrica e ter uma dada localização no espaço.

Recursos – Conceitos que descrevem a utilização de um objecto, essencialmente dentro de um processo.

String – Sequência ordenada de caracteres (símbolos) escolhidos a partir de um conjunto pré-determinado.

Template – Modelo sem conteúdos mas com definições pré-definidas e instruções sobre onde e qual o tipo de conteúdo a incluir no ficheiro.

ABREVIATURAS E SIMBOLOS

ABREVIATURAS

AEC – Arquitectura, Engenharia e Construção

ASCII – *American Standard Code for Information Interchange*

AVAC – Aquecimento, Ventilação e Ar Condicionado

BIM – *Building Information Modeling*

BREP – *Boundary REPresentation*

CAD – *Computer-Aided Design*

CAM – *Computer-Aided Machining*

CIS/2 – CIM*steel Integration Standards*

CSG – *Constructive Solid Geometry*

GUID – *Globally Unique IDentifier*

HTML – *Hyper Text Markup Language*

IAI – *International Alliance for Interoperability*

IDM – *Information Delivery Manuals*

IFC – *Industry Foundation Classes*

IFD – *International Framework for Dictionaries*

ISO - *International Standardization Organization*

MEP – *Mechanical, Electrical, Plumbing*

MSG – *Model Support Group*

MVD – *Model View Definitions*

REBAP – Regulamento de Estruturas de Betão Armado e Pré-esforçado

RSA – Regulamento de Segurança e Acções

SEM – *Semantic Exchange Modules*

STEP – *STandard for the Exchange of Product data*

XML – *eXtensible Markup Language*

SÍMBOLOS

α – Coeficiente de dilatação linear

E – Young's Modulus

ν – Poison's Ratio

G – Shear Modulus

ρ – Density

ζ – Dumping Ratio

1
INTRODUÇÃO

1.1. CONSIDERAÇÕES INICIAIS

Ao longo da história, sabemos pelos monumentos megalíticos, templos faraónicos e pelos grandes impérios que a construção milenar era já vista como um sinal de evolução, que foi escrevendo a história da presença humana ao longo dos séculos. Com a revolução industrial e o fascínio do homem pelos inovadores sistemas mecânicos, esta visão sobre a construção foi-se deteriorando e foi ainda mais agravada com o surgimento da era tecnológica que agora vivemos. Definitivamente a construção tinha passado para um segundo plano, os seus processos de execução e tecnologias mantiveram-se quase que inalterados ao longo das últimas décadas, chegando mesmo a ser considerada como uma indústria de processos rudimentares onde predominava a mão-de-obra pouco qualificada.

Os tempos que correm são já vistos como um ponto de viragem. A indústria da construção apercebeu-se que por si só já não era auto-sustentável em termos de inovação e por isso começou a procurar meios impulsionadores que permitissem de certa forma a recuperação do tempo perdido. Aliando-se às ferramentas e sistemas mecânicos e aos dispositivos e inovações tecnológicos, a indústria da construção encontrou os meios necessários para aprimorar as suas soluções construtivas, facilitar o processo de construção, aperfeiçoar o desempenho dos edifícios e melhorar significativamente os níveis de qualidade de vida do Homem. Estava então estabelecida uma relação de mútuo benefício pela interligação das várias indústrias.

O conceito de *Building Information Modeling* (BIM) surge então como um dos avanços mais promissores na indústria de Arquitectura, Engenharia e Construção (AEC). Este conceito parte de uma filosofia associada à "*Lean Construction*" que se baseia na minimização de gastos supérfluos através da maximização do aproveitamento de recursos. Melhor traduzindo, o conceito BIM fornece uma série de ferramentas que possibilitam a diminuição de tempo e custos de projecto pela globalização de um modelo virtual tridimensional partilhado, melhorando assim a relação de comunicação entre os intervenientes e envolvendo-os mais activamente no projecto reduzindo os tempos de produção. Esse modelo poderá ser usado ao longo do ciclo de vida do edifício, permitindo um maior controlo em todas as fases de construção contribuindo também para a redução de tempos de execução, custos de não qualidade e detecção de não conformidades, promovendo um maior aproveitamento dos recursos já que o planeamento de tarefas é facilitado.

Através das diversas fases do ciclo de vida de um empreendimento muitas são as ferramentas utilizadas e intervenientes envolvidos. A globalização do modelo virtual BIM e a sua partilha entre os diversos intervenientes, é garantida pela interoperabilidade entre sistemas que permite a troca de dados e informações. O IFC (*Industry Foundation Classes*) é o principal formato de troca de dados

responsável pela interoperabilidade entre os diferentes sistemas BIM. Permite o reaproveitamento de modelos já criados evitando a repetição do trabalho de modelação do edifício sempre que se muda de disciplina do projecto, trazendo ainda o benefício de aproximar todos os intervenientes em fase de projecto. Esta aproximação em detrimento da individualização dos processos de trabalho estabelece uma relação colaborativa que trás grande benefício e acréscimo de valor ao projecto (Figura 1.1).

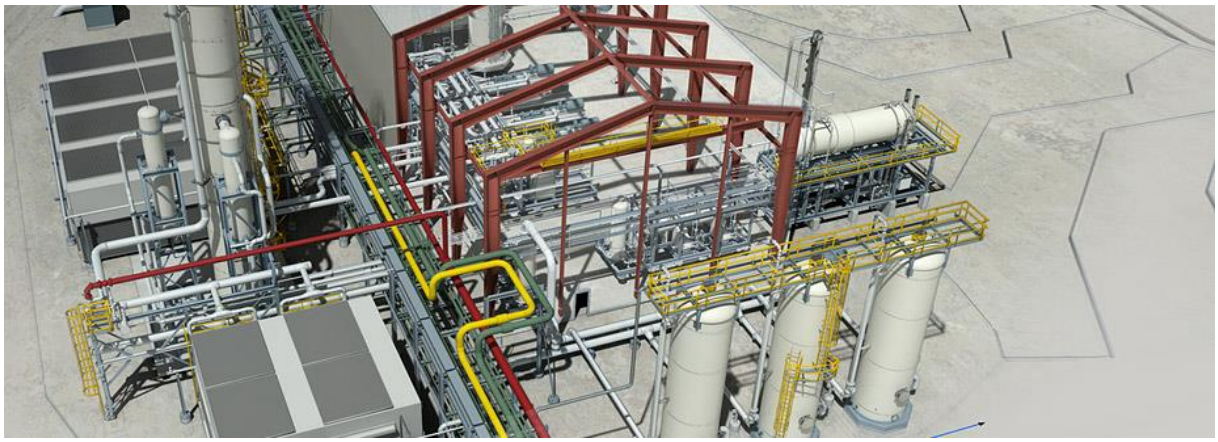


Figura 1.1 – Modelo BIM com a inclusão de várias disciplinas. Adoptado de (Autodesk 2013).

1.2. Âmbito e Objectivos

O presente trabalho foi elaborado no âmbito da disciplina "Dissertação em Construções" no ramo de especialização de Construções do curso de Mestrado Integrado em Engenharia Civil da FEUP.

Apesar de constituir uma preciosa ajuda na transmissão de dados entre diferentes *softwares* que de outra forma não seria possível, é de conhecimento geral que o IFC está ainda longe de ter uma operabilidade perfeita. Desde a sua criação que vêm sido feitos grandes esforços para que a especificação IFC se consiga impor e entranhar num mercado moldado pelas grandes empresas líder. Enquanto formato aberto e com uma filosofia openBIM, o IFC apenas se pode limitar a procurar a aceitação por parte dos distribuidores de *software*, a quem compete libertar as funcionalidades de modo a que possam ser passíveis de utilização pelo modelo IFC. Esta problemática conduziu a um primeiro objectivo que consiste em identificar os principais problemas de interoperabilidade e limitações que o formato IFC apresenta na troca de informações pela realização de testes de interoperabilidade, aqui com especial ênfase para a vertente dos elementos estruturais. As conclusões extraídas servirão principalmente como alerta para as dificuldades na realização de trocas entre programas vocacionados para a modelação e programas vocacionados para a análise estrutural, dando também a conhecer um pouco do modo de funcionamento desses programas que serão avaliados.

Esses testes de interoperabilidade usando ficheiros IFC alertaram para um facto, a dificuldade que há em compreender o próprio esquema IFC é uma barreira que se opõe à interoperabilidade. Tendo isto em conta pretendeu-se colmatar algumas lacunas existentes, como a falta de guias para iniciados no estudo do IFC e a falta de exposição de casos práticos. Portanto, outro dos objectivos desta dissertação foi a compilação de um conjunto de informações sobre BIM e sobre o funcionamento da linguagem IFC que facilitasse o primeiro contacto com estas matérias, visto serem assuntos ainda pouco enraizados não só na comunidade de estudantes mas mesmo até na comunidade de profissionais de engenharia e outras áreas. Houve então uma preocupação adicional em fazer-se uma abordagem a esses assuntos vocacionada especialmente para quem inicia o estudo nestas áreas pela primeira vez.

Este objectivo da exposição do funcionamento da linguagem IFC traz consigo outro propósito. Sabendo-se que há um interesse crescente em mecanizar determinados processos para facilitar actividades, surge o objectivo de pesquisa de conhecimentos adicionais para integrar essa exposição, que depois poderão ser utilizados para o desenvolvimento de ferramentas de geração de dados. De facto, se abstraídos alguns processos do contexto em que se inserem, damos conta que não representam propriamente actos de engenharia, mas apenas actos repetitivos cuja execução é necessária mas que poderia ser feita por um outro individuo com menos qualificações ou até de forma automática. Foi neste domínio que se procurou implementar os objectos estudados, para que de alguma forma se conseguisse diminuir trabalhos de modelação com caracter mais repetitivo, mostrando também assim a aplicabilidade da especificação IFC enquanto ferramenta que proporciona a mecanização de processos, através da criação de uma ferramenta informática que possibilite a modelação de estruturas porticadas pela introdução de informações genéricas (como por exemplo o número de pórticos e distância entre eles, número de pisos, secções, materiais, cargas aplicadas, etc.).

"Interoperability will become more of an issue as we continue to push for the use of new technology, whether it be BIM or any other technology that gets us closer to an integrated project delivery system. Our goal is to be able to develop within the project [build] team an attitude that improves the process by having teams that are fully integrated and cooperative with each other. You need to be able to take advantage of technology that allows the free flow of information from the submittal stage to the operations and maintenance phase. We see interoperability as a challenge to achieving that."

– Ricardo Aparicio, Manager, General Electric President, Construction Users Roundtable

1.3. ORGANIZAÇÃO DA DISSERTAÇÃO

No início da presente dissertação, ainda com os objectivos finais pouco definidos, o título "*O Modelo IFC como Agente de Interoperabilidade – Aplicação ao domínio das estruturas*" sugeria numa primeira fase a investigação e identificação de problemas no uso do IFC na interoperabilidade entre programas e numa segunda fase a elaboração de propostas de aplicação com uma vertente voltada para o domínio das estruturas. Estando este caminho delineado, tornava-se necessário aprofundar o conhecimento, no domínio dos BIM e do IFC. É nesse sentido que surge o "*Capítulo 2 – Estado da Arte*", provém da necessidade de reunir uma série de informação que possibilite reter os conhecimentos necessários para o estudo aprofundado destas matérias.

Com esses conhecimentos de base torna-se possível a compreensão do modo de funcionamento do modelo IFC e da sua estrutura de troca e surgem as primeiras impressões acerca da interoperabilidade desse modelo entre os diversos programas de *software*, resultando no "*Capítulo 3 – Estudo Empírico de Interoperabilidade*" que contempla casos de estudo específicos de transferência de informação entre programas, quer de forma indirecta usando o IFC como intermediário para permitir essa troca de informação, quer por via directa se possível. Com este Capítulo 3, fez-se a identificação e classificação da performance geral do IFC enquanto agente de interoperabilidade, o que responde à primeira fase sugerida pelo título da dissertação. Para além disso, o Capítulo 3 veio alterar o modo de encarar a problemática da questão, neste momento a investigação e preocupação principal já não eram os problemas ligados à interoperabilidade propriamente dita, pois percebeu-se que o maior problema residia na origem, na própria implementação do modelo IFC. Uma correcta implementação do IFC depende em muito da aceitação desse modelo por parte dos principais desenvolvedores de *software* na indústria AEC (Arquitectura, Engenharia e Construção), mas não basta que haja uma aceitação formal, é preciso que esses programas criados sejam capazes de suportar o modelo IFC na sua totalidade. Para

isso é necessário que os distribuidores destes *softwares* estejam dispostos a liberar a informação que é gerada pelos seus sistemas informáticos para a estrutura de troca do IFC, para que possa ser usada por outros programas que podem ou não pertencer ao mesmo distribuidor. Colocando o cerne da questão nestes termos, compreende-se a dificuldade e inércia que existe em implementar esta filosofia do IFC, mas é nesse sentido que vão todos os esforços feitos pela buildingSMART. Este é o verdadeiro objectivo do formato IFC, ser um formato aberto, um verdadeiro openBIM, que tenta por isso fazer a ligação entre todos os programas BIM, não demonstrando especial empatia por nenhum programa específico.

Foi visto que na actual situação, tanto a exportação como a importação de entidades entre programas sofria uma espécie de "triagem" e selecção da informação, já que se verificou que na generalidade a única informação que circulava entre os programas era apenas informação geométrica. A constatação deste problema conduziu à segunda fase da dissertação, que consistia na elaboração de propostas de aplicação com uma vertente voltada para o domínio das estruturas, de uma perspectiva que procurasse explorar as entidades que definissem os elementos da construção, mas desta feita indo mais além do que a definição arquitectónica, aprofundando a definição de forma estrutural. Para esta especial vertente estrutural o esquema IFC2x3 revelou-se insuficiente para uma completa definição dos elementos pretendidos e como o "*timing*" do início da dissertação coincidiu com o da realização do novo esquema IFC4 que contempla uma área de entidades mais abrangente, tomou-se este esquema como o esquema que daria a base ao estudo das entidades, exposto no "*Capítulo 4 – Modelação em IFC (Caso de Estudo)* ". Este capítulo constitui um guia para aqueles que pretendem iniciar o estudo na área das classes IFC, dando indicações do modo de funcionamento do código e linguagem usados pelo IFC, sempre com a preocupação em fazer a exposição da informação de um modo que facilite a compreensão e apreensão dos principais conceitos.

O Capítulo 4 serve também de elo de ligação entre a problemática levantada no Capítulo 3 e a aplicação prática demonstrada no "*Capítulo 5 – Proposta de Implementação de Classes IFC na Criação de Softwares*". O capítulo 5 representa o culminar de todo o conhecimento apreendido na realização da dissertação, sugerindo aplicações de *software* que contribuirão para um aumento dos níveis de interoperabilidade. Pretende-se que o desenvolvimento dessas ferramentas seja feito de forma parcelar, onde dos avanços no estudo das classes IFC sejam retirados os conhecimentos sobre as entidades estudadas de modo que seja possível incrementar funcionalidades nesses *softwares*. Então, ter-se-á uma implementação baseada no conceito de melhoria contínua, cujo objectivo será a incrementação e a procura por soluções que contornem os actuais problemas de interoperabilidade.

Por último surge o "*Capítulo 6 – Conclusões"* que como o próprio título indica, vem tecer as considerações finais com base no estudo elaborado ao longo da dissertação, expor algumas das principais dificuldades sentidas ao longo desse estudo e dar também um ponto de vista mais pessoal acerca da situação actual no que se refere à interoperabilidade entre ferramentas BIM.

2
ESTADO DA ARTE

2.1. GESTÃO DA INFORMAÇÃO

2.1.1. ASPECTOS GERAIS

Uma boa base para a comunicação reside numa gestão de informação apropriada, sendo que os BIM (*Building Information Modeling*) surgem como a ferramenta que possibilita a integração de toda a informação referente a um dado projecto e de todos os intervenientes envolvidos, constituindo assim um dos desenvolvimentos mais promissores na área da arquitectura, engenharia e indústria da construção (Eastman et al. 2011).

O recurso às ferramentas BIM constitui um contributo valioso no alcance dos objectivos do projecto, tornando todo o processo de gestão mais transparente. O modelo tridimensional demonstra rapidamente quais os pontos que requerem especial atenção, quais os objectivos cumpridos em determinada área, quais as situações que futuramente são susceptíveis de evidenciar problemas, etc., resumidamente, apenas pela visualização do modelo 3D é possível traçar cenários expectáveis e definir medidas concretas de correcção. Percebesse então claramente quais os benefícios que a implementação destas ferramentas trás para todas as áreas do projecto, já que permitem uma melhoria dos conceitos básicos da acção e interacção humana (colaboração, comunicação, compreensão e visualização). Contudo, por si só a indústria da construção não se altera apenas com a implementação de novas ferramentas tecnológicas, é fundamental que haja um contributo e um trabalho realizado em equipa por parte dos colaboradores do projecto para que se alcancem os resultados desejados. (Kymmell 2008).

2.1.2. DOCUMENTAÇÃO DE PROJECTO

Nos dias que correm a indústria da construção regula-se por desenhos na sua maioria bidimensionais (peças desenhadas) e instruções escritas (peças escritas), que permitem assim a transposição de ideias para o papel, de modo a que essas informações sejam perceptíveis por toda a comunidade da construção. Nas últimas décadas, inúmeras normas vêm acompanhando a evolução e desenvolvimento dos desenhos de construção e restante documentação de projecto, mas apesar destas normas e especificações técnicas estarem uniformizadas, o uso de instruções bidimensionais num mundo tridimensional pode conduzir a conclusões ambíguas. Este é um problema recorrente na indústria da construção, os baixos níveis de formação de quem executa associado à dificuldade de interpretação das peças de projecto e transposição visual de informação bidimensional para tridimensional, conduz

muitas vezes a uma distorção da ideia conceptual pensada pelo projectista, o que geralmente resulta em erros de execução. Devido à dificuldade que existe em compreender se a informação foi correctamente percepcionada, muitas vezes estes erros de comunicação e interpretação mantêm-se ocultos até que seja demasiado tarde para abordá-los de forma eficaz.

A organização das peças de projecto pode demonstrar ser bastante complexa, pequenas alterações como por exemplo a modificação da localização de uma porta ou janela implicam a necessidade de alteração de várias peças desenhadas e escritas. Deve ser assegurado que todos os documentos que contêm informações relativas à referida porta ou janela são devidamente alterados, porque uma peça não alterada irá levantar conflitos entre versões. Mesmo procedendo a uma correcta alteração de todas as peças existe ainda a dificuldade em fazer chegar essa nova informação a todos os envolvidos na construção. Estes pequenos exemplos servem para salientar a importância da comunicação e realçam as potencialidades dos BIM (Figura 2.1).

Figura 2.1 – Os âmbitos dos BIM e suas inter-relações. Adoptado de SCIA Scientific Software.

A natureza repetitiva da informação contida nos desenhos é também uma das causas para a ocorrência de erros. Esse é um dos aspectos para o qual a ascensão da era informática trouxe bastantes benefícios, a automatização dos processos veio acelerar a elaboração dos projectos e diminuir significativamente os erros associados a esta problemática (Kymmell 2008).

2.1.3. Gestão de Projecto (*Project Management*)

Os objectivos do projecto derivam directamente dos desejos e das necessidades do Dono de Obra e também dos membros da equipa de projecto. Para que se alcancem os objectivos e se obtenham os resultados desejados é necessário transpor as barreiras que se opõem ao desenvolvimento do projecto. Para isso, devem-se conhecer as fraquezas existentes no processo de construção, deve ser incentivada a redução dos riscos, a colaboração entre intervenientes, a antecipação dos problemas e o aumento da segurança mas também nunca esquecendo o controlo de custos e tempo.

"O maior problema no planeamento e construção de projectos de edifícios é a incorrecta visualização da informação de projecto". Certo é que não é possível representar e definir um projecto do qual não se tem uma plena compreensão dos seus requisitos e objectivos. Um projecto deve ser visualizado, compreendido e comunicado de forma completa desde o seu início para que se limite ao máximo a ocorrência de erros a priori.

As principais causas geradoras de fraquezas no planeamento, concepção e processo construtivo são:

- Dificuldades de Comunicação – A complexidade dos projectos de construção e os inúmeros recursos humanos envolvidos são sempre um problema de difícil gestão. Fazer chegar a informação da origem até ao destino de forma intacta é por vezes uma tarefa que exige determinados cuidados, pois muitas vezes a informação é difundida por meios voláteis que distorcem as informações. Para além desses problemas também poderão existir problemas no entendimento das tarefas ou até mesmo problemas de comunicação por incompatibilidade entre o carácter de indivíduos.

- Competição entre membros de uma equipa – Por vezes indivíduos membros de uma equipa colocam os seus interesses pessoais à frente dos interesses da equipa, o que resulta numa competição entre indivíduos com interesses pessoais, quando deveriam estar todos à procura das soluções que melhor respondessem ao interesse da equipa e do projecto.

- Transferência de risco – Na maioria dos casos os vários modelos contratuais apenas transferem o risco de uma equipa para a outra, e por isso no final o Dono de Obra é quem acaba por suportar os custos de ineficiência e dos problemas associados ao projecto. A maioria dos projectos não incentiva a colaboração.

- Litigação – Devido à enorme complexidade existente em todo o processo de construção, são criadas demasiadas situações susceptíveis de gerar desentendimento (podendo-se originar problemáticas relacionadas com a resolução de conflitos, erros e omissões, etc.). A litigação dá-se devido à falta de comunicação, compreensão e colaboração entre os membros da equipa de projecto, portanto quase todas as diferenças conseguem ser trabalhadas através de diálogo e comunicação construtiva. É do interesse da equipa minimizar e evitar estas situações pois apenas terceiros terão benefícios com os gastos causados por este tipo de episódios.

Para combater estes aspectos negativos torna-se necessária a implementação de uma série de medidas correctivas antes do problema ocorrer. Para isso, é necessário antes de mais compreender as fraquezas que estão associadas à construção para que se faça um planeamento e monitorização eficaz desses problemas, que responda às necessidades, seja eficaz e que vá de encontro com metodologias de melhoria continua, dessa forma os objectivos serão mais facilmente alcançados.

Estas são algumas das medidas para prevenção da ocorrência de falhas e também de gestão de conflitos:

- Redução do risco por melhoria da comunicação – Uma comunicação deficiente é um factor crítico para a criação de risco na construção. A complexidade da construção constitui um ambiente propício ao aparecimento de oportunidades onde a comunicação poderá ser feita de forma deficiente e por isso ser mal percebida ou não entendida por completo. Cada membro de uma equipa de projecto deve ter a responsabilidade de assegurar que a comunicação é feita por moldes que garantem a correcta transmissão da informação e que esta é feita de maneira clara e perceptível pelos restantes indivíduos envolvidos num processo de construção.

- Redução do risco por aumento da colaboração – O facto de colaboração entre equipas é também um factor de redução de risco. Geralmente os membros de uma equipa preferem trabalhar de

forma individual, pois desse modo não terão que partilhar quer o seu sucesso quer o seu insucesso com os outros membros. Ora, esta filosofia de pensamento torna o trabalho mais individualista e ao invés de haver uma equipa que procura um objectivo comum, tem-se vários indivíduos com objectivos pessoais que podem não corresponder aos objectivos do projecto, o que faz com que haja dispersão dos trabalhos e um isolamento na comunicação.

- Antecipação dos problemas – A antecipação dos problemas é (quase que) uma das regras de ouro que se procura seguir na construção, mas nem sempre é fácil prever as situações que poderão ser fonte de risco e/ou problemas. Isto implica que haja uma certa capacidade de previsão dos problemas, capacidade essa que pode ser melhorada pelo conhecimento dos factores de risco associados às diversas tarefas e compreensão profunda dos detalhes e processos construtivos de modo a que esses riscos sejam contabilizados e as acções de controlo implementadas ainda na fase de projecto.

- Melhoria da segurança – A segurança deve ser a prioridade em qualquer projecto de construção. Um bom plano de segurança e saúde é uma boa base para garantir a segurança em obra, mas devem ter-se em conta outras acções de prevenção. Os indivíduos envolvidos num projecto devem estar ocorrentes dos perigos e factores de risco, deve existir sinalização abundante em obra, devem ser feitas acções de formação, etc.

- Redução de custos – A redução de custos deve ser feita pelo estudo de mercados paralelos e pela aplicação de filosofias "*lean construction*" que incentivam a eficiência na construção. Os processos de gestão da construção usam ainda pouca tecnologia quando comparados com outras indústrias que têm cada fase do processo completamente monitorizada (Kymmell 2008).

Posto isto, as ferramentas BIM são no fundo uma inovação nos processos tecnológicos do sector da construção que demonstra ter o potencial necessário para ser capaz de mudar não só o modo como a gestão da projecto é feita mas também o modo como é feita a gestão da construção ao longo das fases construtivas e ciclos de vida do edifício. O acompanhamento faseado que o BIM permite fazer, associado ao conhecimento das fraquezas na gestão da construção e medidas preventivas que devem ser tomadas, permitirá revolucionar a indústria da AEC e o modo como o projecto é encarado.

2.2. BIM – BUILDING INFORMATION MODELING

2.2.1. CONCEITO

Um *Building Information Model* é um modelo estrutural de informação digital, tridimensional, onde constam os vários objectos que compõem o edifício, capturando a sua forma, comportamento e relações das várias partes do edifício, sendo possível indexar todo um conjunto de dados a um determinado elemento. Permitem a elaboração de modelos digitais com informação completa de todos os diferentes domínios do projecto, contendo a geométrica e a informação relevante necessária para dar suporte à construção e gestão das actividades necessárias para a elaboração do edifício (Eastman et al. 2011).

A automatização de processos e integração da informação é um dos princípios que está associado ao BIM e uma das suas maiores vantagens. Este conceito é de extrema importância já que permite a implementação de dados externos, como por exemplo, dados provenientes da monitorização de elementos estruturais (Hipólito de Sousa et al. 2011).

Estas primeiras impressões deixam claro que a grande inovação trazida pelo conceito BIM reside em mais do que uma mudança tecnológica, já que representa uma mudança de processos. Ao permitir a representação de edifícios através de elementos inteligentes, capazes de conter informação e relações

entre si, as ferramentas BIM não só mudam o modo como os desenhos de edifícios e respectivas visualizações são criadas, como também mudam todos os processos de elaboração e construção de um edifício, sendo que a maximização das potencialidades das ferramentas BIM dá-se quando se expande a sua utilização a todas as fases de construção do edifício (Figura 2.2) (Eastman et al. 2011).

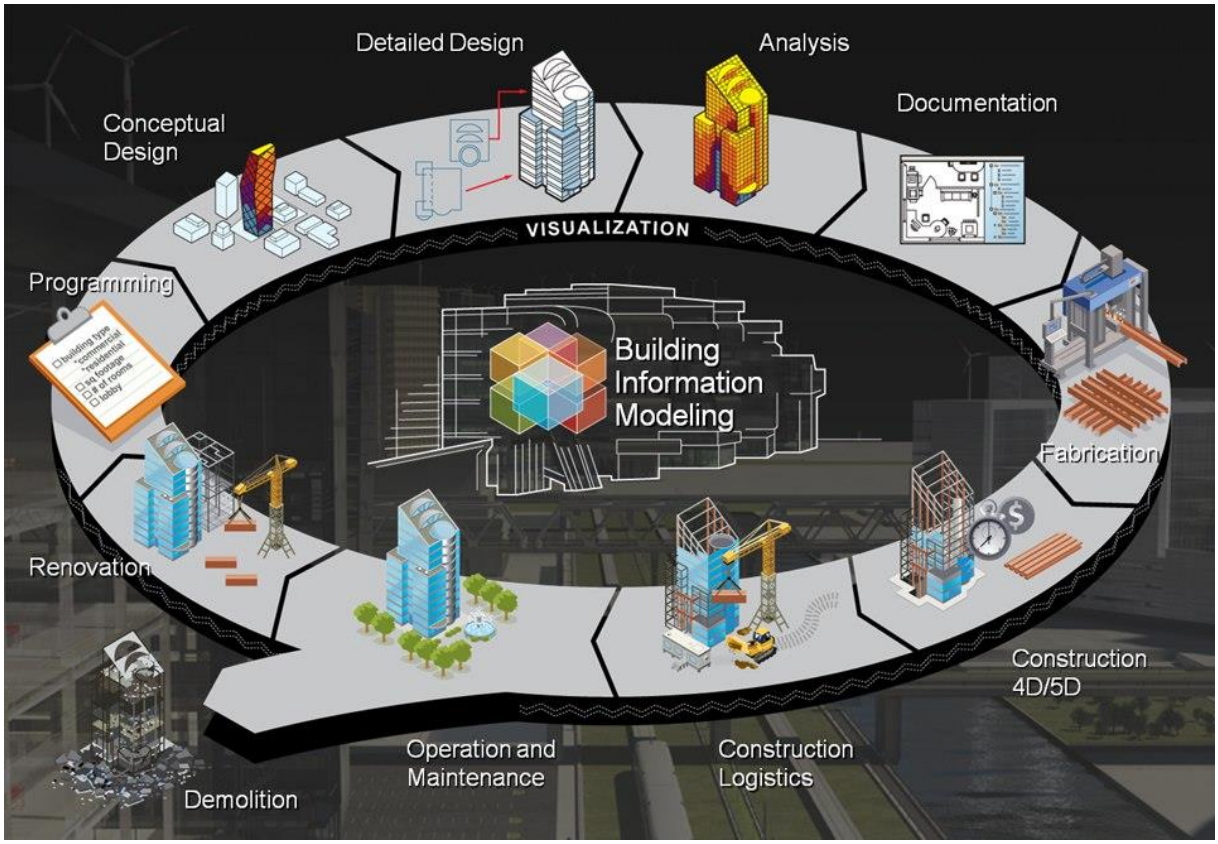


Figura 2.2 – O BIM no ciclo de vida de um edifício. (Autodesk 2013)

Um *Building Information Model* simula o projecto de construção num ambiente virtual. A simulação tem a vantagem de ser feita em computador em formato digital usando um *software* informático, o que permite experimentar as práticas da construção e fazer os devidos ajustamentos antes da produção do documento final. Erros virtuais geralmente não produzem consequências graves se estes forem correctamente identificados e monitorizados para que não sejam transmitidos para o projecto real. O uso das simulações computacionais era já uma realidade para muitas indústrias mas é uma inovação revolucionária no ramo da construção. Quando um projecto é planeado virtualmente é possível visualizar os aspectos mais importantes e fazer a detecção dos erros mais relevantes que podem ser detectados informaticamente antes de se darem quaisquer instruções para o projecto, como por exemplo, a detecção da sobreposição de elementos em fase de estudo usando ferramentas BIM permite uma correcção rápida e eficaz, ainda na fase de concepção de projecto.

Os modelos virtuais geralmente são distinguidos entre duas categorias, modelos de superfície e modelos sólidos. Como o nome indica, os modelos de superfície são apenas representados de forma superficial apenas para efeitos de visualização, através de componentes que contêm apenas informação respeitante a dimensões, forma, localização, etc., ou seja, são modelos ocos que facilitam o estudo de parâmetros visuais do projecto e são usados sobretudo para design estético, planeamento e *marketing*. Já os modelos sólidos, que permitem conter mais informação que os modelos estéticos, são muitas

vezes denominados de "modelos inteligentes" porque possibilitam a simulação em diversos âmbitos da construção, muito para além de apenas simulação gráfica. Um modelo sólido tem ainda a vantagem de permitir produzir modelos bidimensionais como plantas, cortes ou alçados segundo as convenções da documentação da construção (ver Figura 2.3). Isto significa que um modelo sólido pode ser usado numa primeira instância para fazer o desenvolvimento e detalhamento do projecto conceptual e posteriormente esses desenhos poderão ser usados como parte integrante do processo construtivo. Em teoria a elaboração dos modelos BIM já não requer a existência de modelos bidimensionais pois toda a informação pode ser consultada ou retirada do modelo tridimensional, mas na realidade ainda é necessário um progresso destas ferramentas BIM e também uma mudança de mentalidades até que as ferramentas bidimensionais caiam em desuso. (Kymmell 2008)

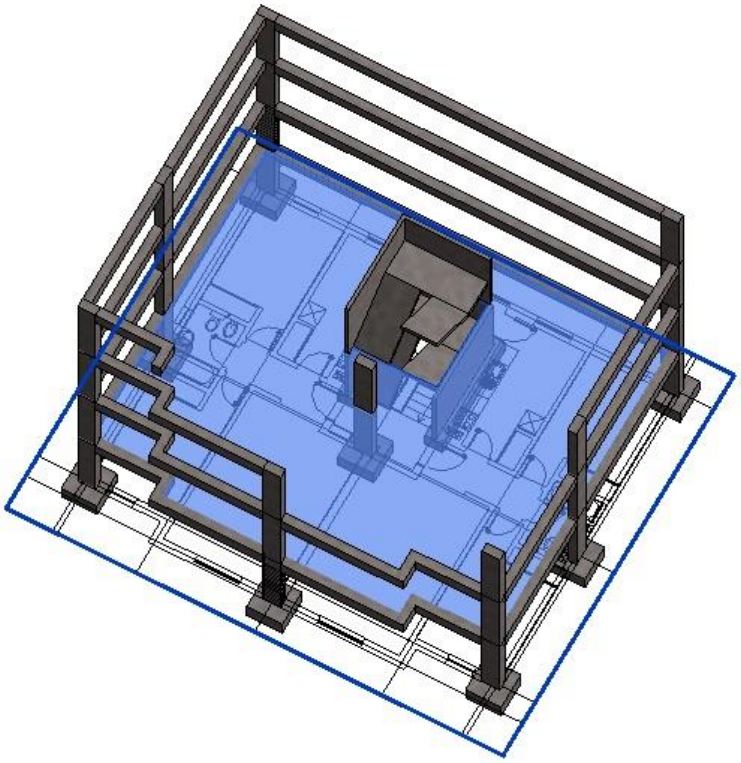


Figura 2.3 – Exemplo de modelação de estrutura 3D a partir uma base CAD

Apesar desse uso ainda não poder ser descartável, começa-se a verificar que está a ser feita a transição do já clássico formato CAD para os modernos formatos BIM e espera-se que num futuro próximos estes modelos sejam usados em 4D (3D + factor tempo) e também em 5D (3D + factor tempo + factor custos). A possibilidade de utilização de plantas, cortes e/ou alçados em formato CAD para utilização de base na construção de modelos tridimensionais BIM constitui muitas vezes uma preciosa ajuda na elaboração destes modelos porque aumentam a facilidade e rapidez de elaboração.

No fundo, os BIM são uma variedade de modelos produzidos por diferentes pessoas, com a possibilidade de diferentes níveis de detalhe e onde está envolto um enorme leque de diferentes *softwares* em formatos distintos. Poderão existir modelos detalhados do terreno (área de implantação do edifício ou infra-estrutura), modelos arquitectónico (o modelo mais usado nos dias de hoje que corresponde à informação geométrica), modelo estrutural (modelo analítico cujo fim será a análise estrutural do edifício), modelos MEP (mecânica, electricidade e canalizações), modelos protecção contrafogo e outros modelos específicos de equipamentos, acabamentos, construção temporária, etc.

Dada a grande variedade de modelos e de utilizadores que poderão estar envolvidos num projecto, a existência de um formato de ficheiros que estabeleça a ligação entre todos esses modelos e utilizadores é uma exigência que se impõe já que será necessário combinar toda a informação num *software* particular, em suma, a interoperabilidade é um ponto fulcral.

Este modelo global do projecto permitirá a realização de análises qualitativas, quantitativa e continua. A análise qualitativa é muitas vezes feita de forma totalmente independente das quantidades, tem essencialmente em consideração a natureza dos problemas. Alguns dos casos são a comunicação, *marketing*, ilustrações e vídeos, que dizem respeito ao conteúdo visual que é elaborado para dar uma impressão acerca do projecto a individualidades externas a este, a análise de construtibilidade, que se refere às metodologias de construção, a coordenação de sistemas e detecção de conflitos, que é uma das aplicações mais populares dos modelos BIM e por fim a análise energética do edifício, que trata da análise qualitativa das soluções construtivas quanto aos ganhos e perdas de energia.

A análise sequencial contempla os estudos nos quais o tempo é um dos factores, tanto em termos de duração como sequência. Esta análise requer um estudo bastante visual mas também inclui muita informação quantitativa. Alguns dos casos são a sequência de montagem e instalação de componentes, que requer uma interpretação visual do modo de execução das tarefas e tecnologias construtivas necessárias e a programação de tarefas da construção que tem de ter em conta a mão-de-obra, equipamentos, meios necessários e todo um conjunto de factores que influência a duração das tarefas.

Por último a análise quantitativa envolve a medição de quantidades de determinado elemento do projecto que muitas vezes é também combinada com outra informação. Alguns dos casos são as quantidades de materiais, que são extraídas do modelo sólido BIM pela análise da informação interpretativa, os custos de construção estimados, que resultam da multiplicação das quantidades de materiais pelos preços de execução respectivos e análise de custos associados à manutenção no ciclo de vida da construção, que contempla os custos de controlo, manutenção e energia.

2.2.2. Vantagens e Desvantagens

2.2.2.1. Vantagens

Para maximização das suas potencialidades, os BIM podem ser utilizados para demonstrar todo o ciclo de vida da construção, para uso presente ou futuro, incluindo os processos construtivos e fases de instalação, podendo melhorar significativamente as práticas comerciais (Eastman et al. 2011). As informações contidas no modelo podem ser utilizadas para os mais variados fins, para a geração de documentos de produção e documentos quantitativos, análises de desempenho de sistemas, análises de incompatibilidades entre outros.

Uma das grandes vantagens apontadas que constitui um dos principais critérios pela escolha de ferramentas BIM é a habilidade de produzir documentos finais de construção (Figura 2.4), sem ser necessário o recurso a outras ferramentas complementares. Segundo Khemlani (2007), as questões relacionadas à integração, ao 4D e ao IFC são ainda consideradas como pouco expressivas. Neste momento, os principais critérios para a escolha de um software BIM são a capacidade de uma produção completa de documentos do edifício, objectos inteligentes que possibilitam uma relação de associação a outros objectos e a disponibilidade das variadas e inúmeras bibliotecas de objectos. A realização de tarefas colaborativas e nomeadamente a capacidade de exportar e importar ficheiros em formatos universais ainda não são critérios significativos para a escolha de ferramentas BIM (Khemlani 2007).

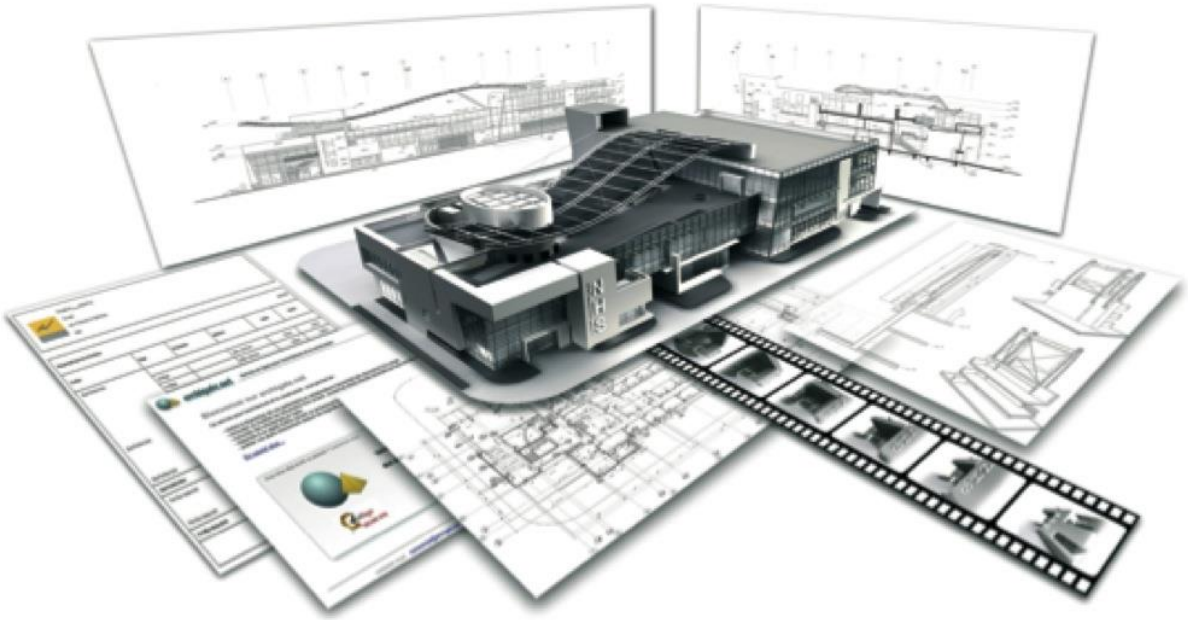


Figura 2.4 – Extracção de informação diversa do modelo 3D (Graphisoft 2013)

Outra das vantagens dos BIM é a possibilidade de trabalho em simultâneo de múltiplas e distintas áreas do projecto, podendo o modelo virtual ser construído por partes, já que os elementos que constituem o modelo estão ligados através de relações paramétricas, possibilitando a introdução da informação de forma estruturada, fazendo com que as alterações se difundam em tempo real em todo o modelo, evitando a propagação de erros e dinamizando os processos de actualização (Eastman et al. 2011). Uma vez que possibilita a avaliação da compatibilização das diferentes peças do projecto significa que há uma maior facilidade na detecção de problemas relacionados com a compatibilização entre os diferentes projectos, permitindo reduzir significativamente a quantidade de erros e omissões de projecto, dando a oportunidade de actuar a priori, ainda em fase de projecto, onde as alterações e correcções são relativamente fáceis de aplicar, evitando possíveis atrasos e gastos em obra, associados a esses factores.

As vantagens de se ter uma equipa de projecto a produzir todo o *Building Information Model* são imensas, porque a resolução de inúmeros problemas é facilitada pela interacção entre indivíduos. Desta forma é possível uma revisão contínua do modelo porque todo a troca de informação é feita numa comunicação constante, resultando num produto final da construção com uma correspondência mais precisa com o que foi inicialmente projectado (Figura 2.5).

O modelo final poderá também permitir a visualização de combinações entre os modelos parcelares, por exemplo, algumas das vistas poderão permitir a visualização do modelo analítico e outra informação relacionada com esses componentes. Existem também outro grupo de ferramentas que permitem a análise quantitativa do projecto, quer a nível de estimativa de prazos, quer a nível de estimativa de custos.

Figura 2.5 – Correspondência entre o modelo BIM e a situação real (Lillja 2012).

2.2.2.2. Desvantagens

Contudo, a utilização das ferramentas BIM ainda são utilizadas por um pequeno número de técnicos, sendo muitas vezes ignoradas as capacidades destas ferramentas já que a maioria apenas as usa para representação tridimensional do modelo do projecto, esquecendo todas as restantes potencialidades ligadas ao conceito (Gequaltec 2011). Os custos de implementação e formação de profissionais são outra desvantagem que dificulta a tomada de decisão pelo seguimento dos BIM.

Muitos aspectos legais relacionados com a implementação das ferramentas BIM têm estado sob discussão e são uma área à qual os arquitectos, engenheiros, empreiteiros e subempreiteiros estão atentos. Existem vários obstáculos à adopção do modelo BIM, como são exemplo as questões relacionadas com a propriedade do modelo e suporte dos custos que lhe são inerentes e funções de encargo pelo rigor e exactidão do modelo. Salvo raras excepções, os contractos correntes não definem concretamente muitos destes aspectos ligados à implementação dos processos BIM e ao não abordarem concretamente os benefícios e riscos associados à introdução destas ferramentas não é possível tirar o máximo partido do seu uso. Contudo e apesar do risco assumido por quem decide implementar o método BIM, esta implementação resulta geralmente numa poupança significativa, razão pela qual o número de empresas que recorrem às ferramentas BIM tem aumentado. Seria vantajoso desenvolver um método de contrato onde os participantes na equipa de projecto pudessem beneficiar das melhorias trazidas pelas técnicas de gestão BIM, promovendo a colaboração e a redução de risco (Kymmell 2008).

2.2.3. Ferramentas BIM

Nas últimas décadas, temos assistido a uma fase de transição em que os desenhos e projectos em suporte papel se têm transposto para o formato digital. A rápida evolução destes instrumentos digitais tem proporcionado inúmeras novas possibilidades ligadas à gestão da informação na indústria da construção (Eastman 1999).

Tudo começa em 1957 quando Patrick J. Hanratty desenvolve o primeiro *software* comercial CAM (*Computer-Aided Machining*). Cinco anos depois, Douglas C. Englebart dá uma visão daquilo que serão as ferramentas arquitectónicas futuras no artigo "*Augmenting Human Intellect*". Em 1963, surge a primeira ferramenta CAD (*Computer-Aided Design*) desenvolvido por Ivan Sutherland nos laboratórios MIT Lincoln, usando uma interface gráfica conhecida como "*Sketchpad*". Os dois

principais métodos de representação de formas, *Constructive Solid Geometry* (CSG) e *boundary representation* (BREP), aparecem entre os anos 70 e os anos 80. Anos mais tarde, uma empresa Húngara, a Graphisoft fundada em 1982, lança o Archicad em 1984 que seria o primeiro *software* BIM no mundo e continua ainda a ser usado mais de 30 anos depois. (Silva 2011).

Desde então muitas outras ferramentas surgiram, que vêm incrementar valor à indústria da construção através da inovação e melhoria contínua. Hoje em dia o uso das ferramentas BIM permitem gerar edificações a partir de maquetes virtuais, onde se pode facilmente e rapidamente extrair um amplo domínio de informações. Permite gerar vistas tridimensionais mas também plantas, cortes, alçados, pormenores entre outras peças de projecto, tudo isto falando apenas no âmbito de geometria mas muito mais pode ser feito. É possível simular detalhes estruturais e fazer verificações segundo normas que validam o modelo analítico e a sua função estrutural e também fazer cálculos de eficiência energética entre outras utilidades que dessa forma fazem uma associação global dos elementos que constituem o modelo. Para além de tudo isto o recurso às ferramentas permite gerar uma quantificação automática com rigor e detalhe que pode ser usada para fins de execução orçamental, controlo de projecto e aprovisionamento de materiais, minimizando a ocorrência de erros nestas áreas pela facilitação de processos.

Nos sistemas BIM, ao modificar-se o projecto tridimensional, todo os restantes conjuntos de informações mencionados são imediatamente alterados e actualizados, não dando espaço à precursão de erros resultantes de alterações efectuada em fase de projecto. Esta é outra das vantagens que vem impulsionando a implementação do BIM.

2.3. INTEROPERABILIDADE

No âmbito das TIC (tecnologias da informação e comunicação) entende-se por interoperabilidade entenda-se a capacidade de gerir, trocar e comunicar produtos e informação electrónica por via computacional, podendo, por exemplo, tratar-se de troca de informações referentes a dados de projecto entre empresas ou profissionais. Por esta razão a interoperabilidade também é vista como um meio que permite a integração na execução do projecto pois constitui uma implementação para administrar relações colaborativas entre os membros dos vários domínios da construção.

Tal como em muitas outras indústrias, na indústria AEC (Arquitectura, Engenharia e Construção) há uma enorme necessidade de comunicação e consequente troca de dados e informação entre arquitectos, engenheiros, consultores, empreiteiros, subempreiteiros, fornecedores e proprietários, pois o processo de projecto envolve muitos participantes nas diferentes fases. À partida esta comunicação só se conseguiria se todos os intervenientes usassem os mesmos softwares. Como seria expectável, cada interveniente usa o software que melhor se enquadra com os seus objectivos profissionais. Isto implica que para haver uma correcta permuta de informações, devem antes de mais existir garantias de interoperabilidade entre softwares, para que os sistemas possam trabalhar em conjunto e todos os intervenientes falem numa mesma linguagem, transpondo as barreiras que se opõem à comunicação (AIA 2009).

No processo de concepção tradicional de edifícios existe uma perda de valor na passagem de informação entre as fases do processo construtivo (retratada na Figura 2.6) e de forma semelhante há também uma perda de valor na troca de informação entre especialidades do projecto, pois nos moldes clássicos quando um membro conclui a sua fase de trabalho e outro membro inicia a fase de trabalho seguinte, este membro irá fazer uma verificação prévia do trabalho do primeiro e por isso terá de voltar a fazer ou verificar um trabalho que à partida seria desnecessário e por isso tem-se uma perda de

valor do trabalho do primeiro membro. Nos moldes actuais as ferramentas BIM já possibilitam a realização de um projecto a partir de um modelo de base comum a todos os membros envolvidos no projecto de um edifício, o que é proporcionado em boa parte pela interoperabilidade. Desse modo a filosofia BIM contribuirá significativamente para a redução dessas perdas de valor, quer pela facilidade de troca de informações, quer pelo aumento da comunicação entre intervenientes e dessa forma o decorrer das fases da construção aproximam-se da situação ideal. Com os elementos de uma equipa a trabalharem com elevada proximidade durante as fases de projecto estes terão uma maior facilidade em identificar erros nas diferentes vertentes do projecto numa fase onde a correcção desses erros é ainda de relativa facilidade e com isso evitam-se os impactos negativos que esses erros teriam se se propagassem até à fase de construção (resultaria não só em custos acrescidos, mas também em perdas de tempo e potencialmente de segurança e outros).

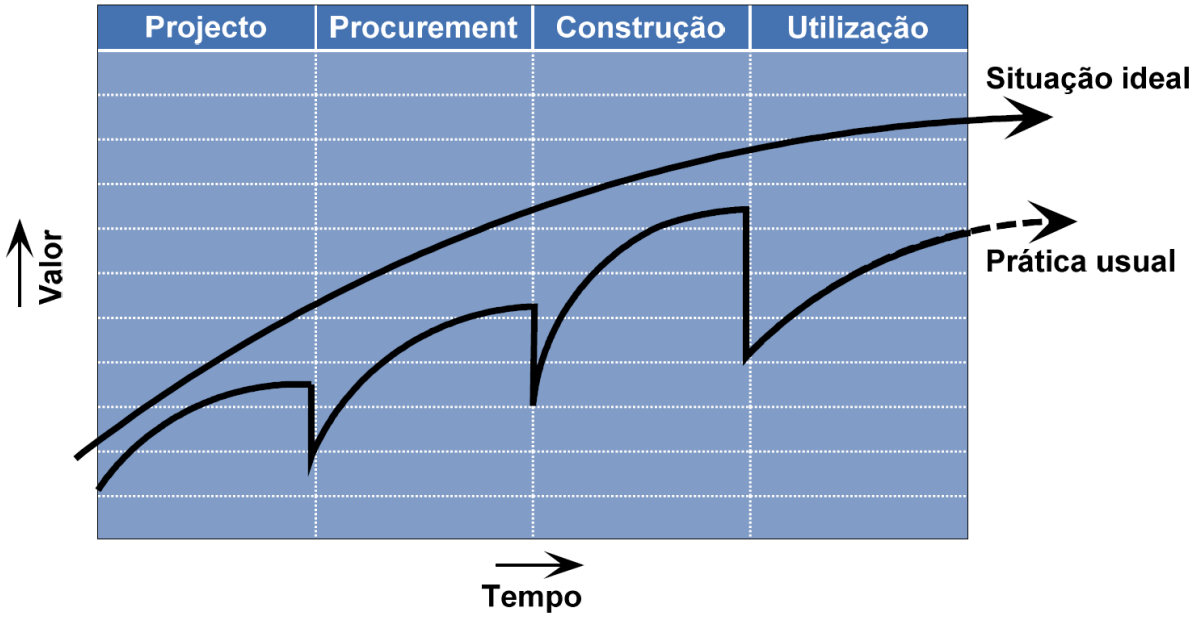


Figura 2.6 – Fragmentação do processo construtivo e suas consequências. (Hipólito Sousa 2006).

Portanto conclui-se rapidamente que todos os softwares envolvidos na indústria AEC devem ter um mesmo objectivo comum, o de facilitar as várias fases e ciclo de vida de um determinado empreendimento. Para isso, estes softwares devem suportar os formatos standard, de modo a permitir a livre circulação de informações e compartilha de dados entre si. Se todos os membros de uma equipa de projecto forem capazes de trocar impressões e informação livremente então poderão fazer uma melhor integração no projecto, resultando num maior rigor e rapidez de processos que aumenta significativamente a eficiência da concretização do projecto a todos os níveis. (Bernstein, Jr., and Jones 2007).

Uma interoperabilidade deficiente tem impactos directos nos custos globais dos projectos de construção e chega a ser apontada como uma das principais causas de desperdício de tempo na partilha e tradução dos dados. Um estudo realizado em 2004 pela National Institute of Standards and Technology aponta a deficiente troca de informações e dados como uma das principais causas dos custos excedentes, estimando uma perda anual associada a este problema na ordem dos 15,8 mil milhões de dólares (USD). Também o relatório sobre interoperabilidade de 2007 da McGraw Hill SmartMarket afirma que, em média, 3,1% do custo de cada projecto é desperdício devido à falta de

interoperabilidade. Estes dados transmitem a problemática associada à deficiente comunicação e a importância de implementação de metodologias mais eficientes para as diferentes áreas (AIA 2009).

As razões que fomentam esta problemática estão relacionadas com a dificuldade associada à transferência de dados entre programas, quer pela vasta quantidade e variedade de informação que um único projecto pode conter quer pela ausência de linguagem comum, que se deve em parte ao interesse que as empresas distribuidoras de software têm, em ocultar determinada codificação própria e privada, de modo que essa informação não seja acessível a empresas da concorrência.

Nos últimos anos, tem existido um esforço que vai no sentido de criar uma linguagem comum para a construção, que visa dar resposta a problemas concretos da construção, relacionados com as organizações, os regulamentos, e as práticas de trabalho actuais. Infelizmente, não é só nestes campos em que há uma falta de uniformização da linguagem. Entre os próprios softwares, denota-se uma ausência de linguagem comum, o que implica a necessidade de reintrodução de dados. A padronização da informação é um desafio a nível técnico e organizacional a ser superado e assim sendo, com o objectivo de suprimir esta lacuna, surgiu como resposta o conceito de interoperabilidade, que retracta a necessidade de transferir informações entre aplicações, permitindo que múltiplos programas possam contribuir para um mesmo trabalho (Eastman et al. 2011).

Neste momento os dois principais modelos de informação de produtos da construção standard, que possibilitam a permuta de informação entre diferentes aplicações BIM são o modelo IFC (*Industry Foundation Classes*) usado para o planeamento da construção, projecto, construção e operação, e o modelo CIS/2 (*CIM steel Integration Standard Version 2*) usado para a engenharia e fabricação de estruturas metálicas. Ambos usam a representação geométrica, relações entre processos e materiais, performance, fabricação e outras propriedades necessárias para o projecto e produção, usando a linguagem EXPRESS também usada em STEP (Standard for the Exchange of Product Model Data) (Eastman et al. 2011).

O objectivo do STEP baseia-se em fornecer o mecanismo necessário para se descrever definições de produtos de uma forma completa e clara ao longo do seu ciclo de vida. Esses dados devem por isso ser elaborados de forma independente do sistema computacional, isto é, as definições descritas devem permitir que a representação da informação dos produtos seja passível de troca entre diferentes sistemas (Laakso and Kiviniemi 2012). O STEP encontra-se organizado numa série de partes em que cada uma se encontra publicada separadamente. Essas partes dividem-se essencialmente em seis grupos que são eles a descrição metodológica, recursos integrados, protocolos de aplicações, testes abstractos, métodos de implementação e testes de conformidade (Schevers and Drogemuller 2006). Como já foi referido o STEP usa formalmente a linguagem EXPRESS para especificar e representar informação de produtos de forma precisa e consistente que facilita a sua implementação. A linguagem EXPRESS inclui tipos, entidades declaradas, contantes declaradas, restrições especificadas e descrições algorítmicas. É uma linguagem de modelação mas não de programação e tornou-se no mecanismo central de suporte para os produtos de modelação abrangendo um vasto conjunto de indústrias, o que só é possível graças ao enorme número de bibliotecas de recursos geométricos, classificações, medições entre outros. A linguagem EXPRESS é de grande utilidade computacional, porém torna-se algo complicada em ser humanamente interpretável e por isso ter sido desenvolvido uma versão mais gráfica conhecida como EXPRESS-G sendo toda essa informação ISO-STEP disponibilizada em domínio público (SCRA 2006).

2.4. IFC – Industry Foundation Classes

2.4.1. Contexto Histórico

A BuildingSMART Alliance surge em 2006, anteriormente denominada *International Alliance for Interoperability* (IAI), é uma organização que nasce da aliança de organizações da indústria AEC cuja missão *"é contribuir para um ambiente construído sustentável, através de informação mais inteligente partilha e comunicação"*. Fundada em 1995, visa desde então criar um padrão que permita a interoperabilidade de dados entre programas na área dos BIM numa indústria da construção que é vista como distribuída e fragmentada dada as várias especialidades que a compõem (como são exemplo as especialidades de arquitectura, estruturas, *procurement*, construção, manutenção e operação). Esse padrão IFC surge com o objectivo de facilitar a transferência de dados representativos de partes de edifícios e suas relações. (Bentley Systems 2008). Desde então têm surgido várias implementações à especificação IFC original. Os principais lançamentos foram o IFC1.5.1, IFC2.0, IFC2x, IFC2x2, IFC2x3 e agora o IFC4. O lançamento do IFC2x em 2000 foi um marco histórico na medida em que a partir desse momento o núcleo da especificação IFC permanece inalterado e cada lançamento adiciona capacidades à especificação anterior.

A Figura 2.7 faz a representação cronológica dos principais avanços do padrão IFC.

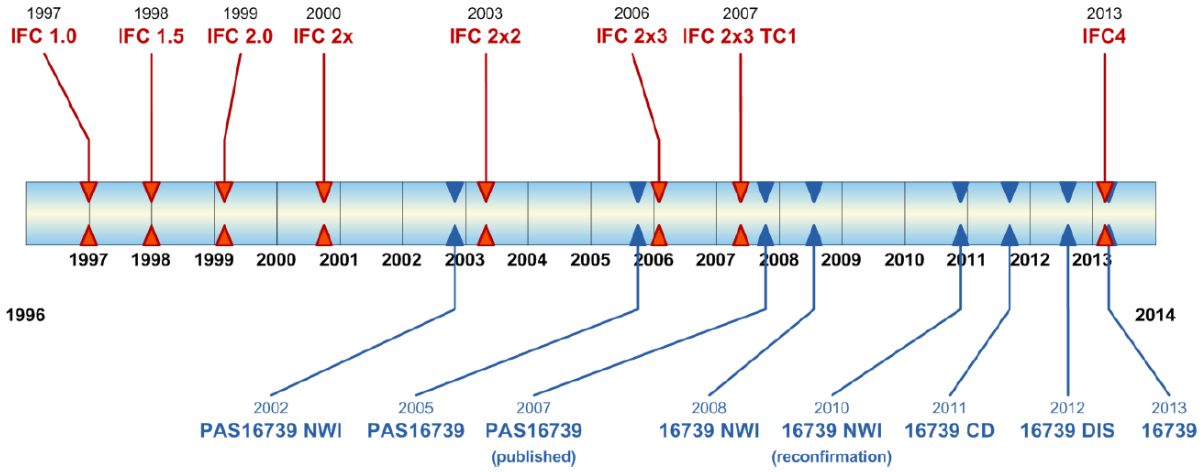


Figura 2.7 – Cronologia do padrão IFC. (Liebich 2013)

2.4.2. Conceito

O principal objectivo desta iniciativa foi o de melhorar a comunicação, produtividade, tempo de entrega, custos e qualidade durante todo o ciclo de vida do edifício, através da possibilidade de permuta de informação entre os intervenientes no projecto independentemente dos programas usados por cada um. Estas são as razões pela qual este padrão desenvolvido é já considerado um pré-requisito para melhorar os fluxos de trabalho na construção usando as metodologias BIM, contribuindo significativamente para eliminar desperdícios gerados por uma interoperabilidade inadequada.

O formato de ficheiros independente, IFC (*Industry Foundation Classes*), é um formato aberto que actua de forma neutra, permitindo capturar não só a geometria mas também todas as outras propriedades associadas aos objectos e também as suas relações dentro de um modelo BIM, para permitir e facilitar o processo de troca de informações que de outro modo não seria possível realizar (Thein 2011).

O IFC é uma base de dados de informação dos objectos que permite a partilha e troca de dados entre softwares, em que cada versão do protocolo IFC permite uma especificação segundo o protocolo ifcXML ou segundo o protocolo EXPRESS (Martins 2012). Usa um arquivo de texto simples, já que este formato é o único formato de dados digital verdadeiramente universal, dando origem a ficheiros com a extensão "*.txt", "*.ifc" ou "*.ifcXML". Note-se que um ficheiro com a extensão "*.ifcXML" tem um tamanho no disco entre 300% a 400% superior a um ficheiro do tipo "*.ifc". Usando um algoritmo de compressão "PKzip 2.04g" também é possível gerar um ficheiro com a extensão "*.ifcZIP", que ocupará entre 60% a 80% do ficheiro original comprimindo ficheiros "*.ifc" ou 90% a 95% comprimindo ficheiro "*.ifcXML" (Bentley Systems 2008).

As aplicações compatíveis podem abrir ou importar arquivos IFC, isto é, podem reutilizar dados gerados em outros programas, adicionar, eliminar ou alterar informação e voltar a exportar novamente para IFC o modelo modificado, permitindo uma sucessiva reutilização desse modelo em outras aplicações que lhe poderão acrescentar dados da mesma ou de outras especialidades. Os sistemas BIM modernos são então capazes de gerar representações ricas em componentes de construção, onde o IFC permite acrescentar uma linguagem comum para a transferência de informações e deste modo as vantagens da exploração dos BIM são maximizadas (AIA 2009). Isto reduz a necessidade de remodelar o edifício na utilização de diferentes aplicações, facto que é incontornável pois o uso produtivo dos BIM requer troca de dados entre as diferentes disciplinas. Cada implementação para a troca de dados usando o IFC deve seguir as exigências de troca. Estas exigências especificam as informações que são necessárias para a troca e partilha de informação numa determinada fase do projecto, sendo esta especificidade de grande importância para prevenir incertezas e certificar e averiguar se a informação em causa foi salvaguardada (BuildingSMART 2013).

"The ability of an owner to reuse data instead of paying to continually recreate or reformat information for new uses during the facility lifecycle is one of most compelling reasons for owners to advocate for true interoperability: a common international data exchange standard that enables data to be imported and exported by any software system, at any time."

- Renee Tietjen, Senior Architect, U.S. Department of Veterans Affairs

2.4.3. Normas

O formato IFC era já registado pela ISO como ISO/PAS 16739:2005 e agora com a chegada do novo esquema IFC4, também ele aceite como norma ISO, vem trazer a esta norma o estatuto de Norma Internacional, constituindo um importante marco histórico no percurso que vem sido feito para tornar o IFC num verdadeiro padrão aberto de BIM. Esta nova norma ISO 16739:2013 vem especificar um esquema conceptual de dados e um formato de ficheiros para a realização de troca de dados entre aplicações BIM, esquema esse que é definido através de uma linguagem EXPRESS, também ela estando normalizada pela ISO 10303-11:2004 e já que faz parte de um formato ASCII (*American Standard Code for Information Interchange*) usa como suporte uma codificação em texto como já havia sido referido. (ISO 2013)

Importa novamente referir que a linguagem EXPRESS usada em IFC é a mesma linguagem de definição de dados que é usada, por exemplo, em STEP ou CIS/2. Tem a vantagem de ser bem adaptada e compacta para incluir regras de validação de dados dentro da especificação desses dados. A estrutura de troca de dados do IFC é chamada de formato "ficheiro físico STEP" e como tal, também é definida pela ISO 10303-21:2002. Desde a realização do IFC2x que a especificação ifcXML faz parte

do IFC-EXPRESS, onde essa especificação é fornecida como um esquema XML 1.0, como definido pelo W3C. O esquema XML é criado automaticamente a partir da fonte IFC-EXPRESS usando a representação XML para esquemas e dados EXPRESS, como definido na norma ISO 10303-28:2007, o que garante a possibilidade de conversão de arquivos ".ifc" para ".ifcXML" e vice-versa, dando consistência à relação entre o IFC-EXPRESS e o ifcXML já que ambos podem trabalhar os mesmos dados (BuildingSMART 2013).

A Figura 2.8 apresenta um diagrama com as normas implementadas pela BuildingSMART para servirem de suporte ao formato IFC.

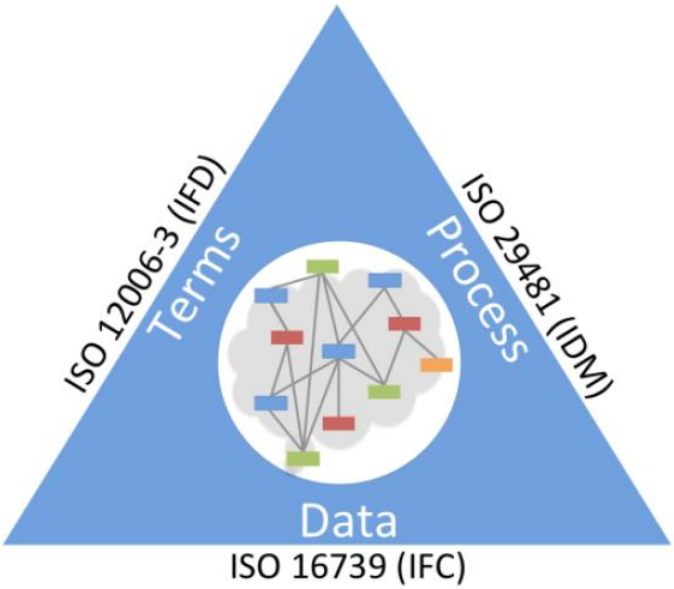


Figura 2.8 – Diagrama com as normas implementadas pela BuildingSMART. (BuildingSMART 2013).

Em 2007 foi introduzido o conceito de *Information Delivery Manuals* (IDM) com a norma ISO 12006-3, como uma especificação que constitui elemento oficial da normalização IFC, foi um dos resultados de uma visão minimalista que conduziu a uma aproximação da normalização da especificação IFC. O IDM tinha duas componentes, uma a de servir as necessidades de implementação a nível técnico para os desenvolvedores de *software* e a outra componente era fornecer os processos que permitissem um fluxo de trabalho entre os utilizadores finais. Um IDM é no fundo uma referência para a integração de dados e processos requeridos pelos BIM e por isso deve especificar o enquadramentos desses dados e processos, porque são relevantes, quem são os envolvidos no processo de criação, qual o tipo de informação em causa, os benefícios da informação e porque é que estas informações devem ser suportadas pelas aplicações. Assim as partes que compõem um IDM são as definições de caracter técnico como o processo de mapeamento, requisitos de troca, partes funcionais e conceitos. (Kiviniemi et al. 2008).

Outra aproximação minimalista para a normalização foi a introdução do formato *Model View Definition* (MVD) do IFC. O seu propósito foi estabelecer um equilíbrio entre as necessidades dos utilizadores enquanto clientes e as possibilidades ao alcance dos desenvolvedores de *software*, através de documentação que expõe o modo como a troca de dados entre diferentes aplicações deve ser feita, descrevendo os benefícios directos da implementação do modelo IFC.

Os conceitos IDM e MVD pretendem facilitar a implementação da interoperabilidade e relacionam-se num contexto que tem como base o IFC. As aplicações implementam um MVD específico que se sustenta depois na documentação fornecida pelo IDM que serve como guia pra o conhecimento dos fluxos de informação permitidos usando a especificação IFC.

Em 2008, a BuildingSMART apresenta formalmente o conceito de *International Framework for Dictionaries* (IFD) que vem alargar o âmbito da normalização do IFC. Este conceito IFD é referido como o "terceiro pilar para a troca de dados baseado no formato IFC" e procura descrever o que é passível de troca facultando os mecanismos que permitem a criação de dicionários para cruzar os modelos IFC com informação existentes em bases de dados. (Laakso and Kiviniemi 2012).

2.4.4. Estrutura do IFC

A especificação IFC é então constituída por definições semânticas informaticamente legíveis e também informação humanamente interpretáveis. Um esquema IFC nomeia relações entre conjuntos de tipos de informação segundo às definições semânticas, podendo esses dados serem verificados em conformidade com um determinado esquema, de modo a garantir que essa representação dos dados está sintacticamente correcta. Tal como outros modelos de dados escritos em EXPRESS, o esquema IFC poderá ser escrito em forma longa ou curta, mas se não for especificada nenhuma forma em concreto significa que o esquema se refere à forma longa. A forma longa comporta todos os tipos de dados num único esquema e a forma curta é usada para fins de documentação e desenvolvimento de esquemas.

A organização do IFC é suportada por uma base de definições EXPRESS que são genéricas para todos os tipos de produtos e definem uma base de construção reutilizável como por exemplo geometria, materiais, propriedades, topologias, medições, intervenientes, cenários e apresentações, que por sua vez são compostos para definir objectos que são usados frequentemente na indústria AEC (Eastman et al. 2011). Os principais objectivos do desenvolvimento do esquema IFC segundo uma arquitectura são fornecer uma estrutura para o funcionamento do modelo, promover um ambiente de partilha e troca de informações entre as vertentes distintas da indústria da construção, facilitar a manutenção da especificação, possibilitar a reutilização de um modelo, permitir a reutilização de componentes de modelação por parte de distribuidores de *software* e facilitar a compatibilidade entre diferentes sistemas. Essa arquitectura opera com base num princípio de escada (Figura 2.9), possibilitando que uma classe refira uma outra na mesma camada ou numa camada inferior, mas nunca numa camada superior. Então as classes de Recursos podem apenas referenciar outras classes de Recursos, as classes de Núcleo podem referenciar classes de Recursos e as Extensões do Núcleo podem referenciar também o Kernel (mas o inverso já não é permitido pelo principio de escada), as classes de Elementos Partilhados podem referenciar classes no Núcleo ou na camada de Recursos e por fim as classes na camada de Domínios podem referenciar quaisquer classes.

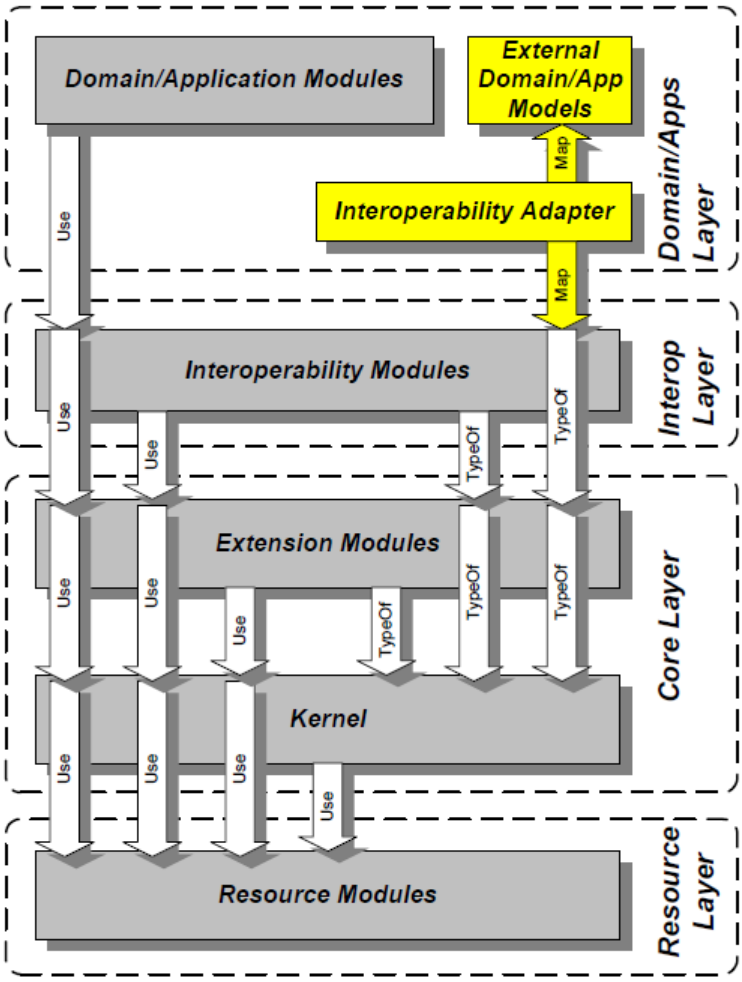


Figura 2.9 – Funcionamento do conceito de camadas na arquitectura do IFC. Adoptado de (IAI 1999).

A constituição de um dado esquema IFC é feita por camadas (sub-esquemas) e segue uma estrutura que está identificada na Figura 2.10. As camadas de Recursos (*Resources*) e do Núcleo (*Core*) dispõem de entidades que possibilitam a definição de modelos especificados nas camadas de Elementos Partilhados (*Shared Elements*) e Domínios (*Domains*).

A camada de Domínios é a camada mais geral que identifica o âmbito de uma especificação IFC. Divide a indústria AEC por grupos de especialidades, dando total independência às entidades pertencentes a esta camada que não pode ser referenciada por uma outra. Um modelo IFC harmonizado é directamente ligado ao núcleo da especificação, já os modelos que não estejam totalmente harmonizados necessitam de definições próprias que garantam o funcionamento através de adaptadores.

A camada de Elementos Partilhados é a seção que diz respeito às questões ligadas com a interoperabilidade entre sistemas. Este nível compreende as entidades que são mencionadas em múltiplos módulos da camada de Domínios. O conceito de módulos comuns permite estabelecer ligações de interoperabilidade entre diferentes domínios e aplicações de modelos. (IAI 1999).

O Núcleo é a camada mais geral dentro de um esquema IFC. Comporta as funções de sustentação do modelo IFC e proporciona a sua estrutura básica, bem como relações fundamentais e conceitos

comuns para especialização em modelos específicos. O *IfcKernel* define a parte central do esquema, capturando a construção em geral e especificando os atributos e relações básicas, como são exemplo a localização de produtos no espaço, a sequência de processos e o agrupamento geral de mecanismos. Todas as entidades fora da camada de Recursos, definidas quer na camada de Núcleo ou acima desta, derivam da classe *IfcRoot*. O *IfcRoot* é a classe mais abstracta e raiz de todas as definições de entidades, ou seja, é o supertipo comum a todas as entidades IFC além das definidas na camada de Recursos. Todas as entidades subtipo do *IfcRoot* podem ser usadas de forma independente enquanto as entidades do esquema de Recursos não são definidas de forma independente. Portanto, estas definições, relações e propriedades de objectos baseadas no formato EXPRESS dividem as entidades em classes de raiz e classes sem raiz, isto é, as classes que derivam directamente do *IfcRoot* são constituídas por um *Globally Unique Identifier* (GUID) que identifica o objecto independentemente da aplicação que está a usar o modelo, um nome, uma descrição e definição de utilizador, em oposição as classes que não derivam da raiz não têm por isso identidade, ou seja, só são definidas se referidas por atributo directo, inverso ou derivado requerido por uma entidade raiz. A decomposição da raiz *IfcRoot* em tipos de objectos de geometria, relações e propriedades é feita por três tipos fundamentais de entidades fundamentais, *IfcObjectDefinition, IfcRelationship* e *IfcPropertyDefinition*, que formam o primeiro nível de especialização dentro da hierarquia. As definições de objectos são a generalização de qualquer item tratado semanticamente dentro o modelo IFC, representando todos os elementos tangíveis fisicamente (como é o caso de sapatas, pilares, vigas, lajes, etc.), items que existem fisicamente (como por exemplo espaços, grelhas, limites virtuais, anotações, linhas de chamada, etc.) e comportando também processos (tarefas, controlo de custos, controlo de recursos, intervenientes, etc.). As relações fazem a generalização do relacionamento entre objectos e diferentes entidades que permite assim especificar relações entre objectos e propriedades directas e desagregar relações semânticas feitas pelos atributos dos objectos. As propriedades definem a generalização da atribuição de características aos objectos que especificam informações particulares e atributos individuais mas que podem também ser partilhadas por múltiplas entidades. Cada um destes três tipos de entidades fundamentais ramificam-se em subtipos de entidades que melhor especificam cada uma das categorias e que constituem o segundo nível de especialização da hierarquia. As definições de objectos decompõem-se em produtos, processos, controlos, recursos, intervenientes e grupos. As relações decompõem-se em atribuições, associação, decomposição, definições, conectividade e declaração. Quanto às propriedades são distinguidas entre conjunto de propriedades de modelo que "definem os tipos de sintaxe e dados para conjuntos de propriedades e propriedades individuais" e conjunto de propriedades de ocorrência que "definem as propriedades partilháveis e extensíveis que podem ser atribuídas aos objectos e ocorrências". (BuildingSMART 2013)

Por último tem-se a camada de Recursos que é a camada onde se encontram as disposições de entidades que suportam as estruturas de dados do modelo. É a camada mais baixa na arquitectura da especificação IFC. Aqui são definidas todas as entidades genéricas que permitem a formulação de um conjunto de dados e informações que possibilitam a construção de modelos, sendo referenciadas pelas entidades raiz de um determinado modelo. Os recursos podem ser caracterizados como sendo um conjunto de entidades que completam todas as outras classes. (Eastman et al. 2011).

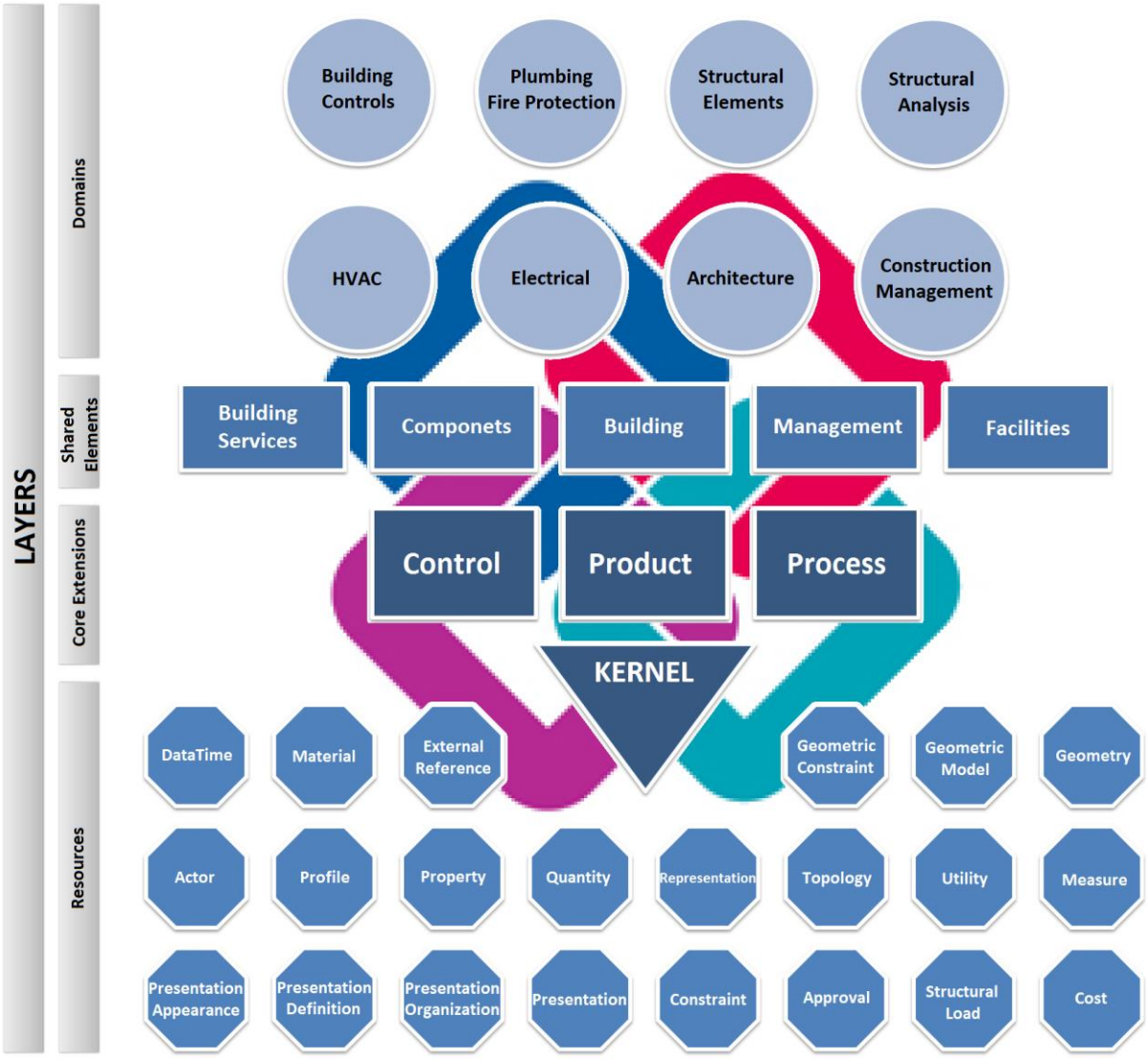


Figura 2.10 – Estrutura da base de dados da especificação IFC4. Adaptado de (BuildingSMART 2013).

Como a especificação IFC é de carácter expansivo, as entidades base podem ser especializadas pela introdução de subtipos de entidades para usos particulares. Esta estrutura hierarquizada de entidades supertipo e entidades subtipo é feita de forma indentada numa árvore hierárquica de relações. Quando um modelo num dado software é traduzido para o modelo IFC usando a relação entre categorias de objectos e entidades respectivas, o modelo em questão é decomposto em tipos de objectos de geometria, relações e propriedades que são dados identificáveis genericamente, para que dessa forma possa ser transposto do formato inicial para o formato IFC. Além destes tipos de objectos a especificação IFC inclui também processos para a representação de actividades usadas na construção de edifícios, na análise geométrica, e análise de dados introduzidos e resultados.

No fundo o IFC consiste numa biblioteca de definições e propriedades que podem ser usadas para representar um projecto de um edifício. Traduzindo o esquema IFC4 em números pode dizer-se que é constituído por 766 entidades, 126 tipos definidos, 206 enumerações, 59 tipos de selecções, 42 funções, 408 *property sets*, 91 *quantity sets* e 1691 propriedades individuais. Estes números

representam a vasta dimensão semântica do IFC e demonstram a complexidade desta especificação que procura cada vez mais possibilitar a tradução de uma área mais alargada de dados e informações respeitantes a edifícios e seus sistemas. Estas listagens organizadas alfabeticamente onde constam todas as informações necessárias para a elaboração de modelos IFC são fornecidas pela BuildingSMART em formato HTML A hierarquização em árvore dessas listagens permite uma navegação e visualização de todos os subtipos de entidades.

2.4.5. O IFC NA INDÚSTRIA AEC

Em suma, o modelo de informação IFC é visto como uma das chaves que permite transpor barreiras e ineficiências que se opõem ao desenvolvimento das tecnologias na indústria AEC (Thein 2011). Consiste numa biblioteca de objectos e propriedades que podem ser usados para representar projectos de edifícios e informação de suporte para um caso particular (Eastman et al. 2011).

Esta realidade só foi possível pela colaboração conjunta de arquitectos, engenheiros, empreiteiros, proprietários de edifícios, técnicos de instalações, gerentes, fabricantes, fornecedores de software, agências governamentais, laboratórios de pesquisa, universidades entre outras entidades (Figura 2.11) que agora contribuem e fazem parte da BuildingSMART (AIA 2009).

A BuildingSMART Internacional é organizada por alianças regionais que representam um país ou grupo de países. As alianças actuais servem a Austrália, Benelux, China, Itália, Japão, Coreia, Oriente Médio, América do Norte, Singapura, Reino Unido, Irlanda, os países nórdicos (Dinamarca, Finlândia, Noruega e Suécia), os países de língua alemã (Alemanha, Áustria e Suíça), os países de língua francesa e os países ibéricos (Portugal e Espanha). (Thein 2011).

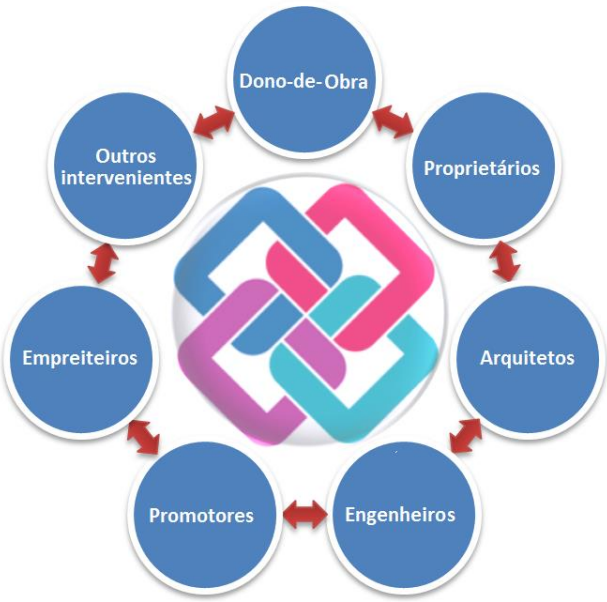


Figura 2.11 – O formato IFC enquanto agente activo de interoperabilidade no openBIM

3

ESTUDO EMPIRICO DE INTEROPERABILIDADE

3.1. INTRODUÇÃO

Com o intuito de se diagnosticar a eficiência e níveis de interoperabilidade dos principais aplicativos comerciais BIM, o presente capítulo representa um esforço feito no sentido de identificar as principais não conformidades pela troca de informações de modelos genéricos. Serão elaborados casos de estudo de transferência de dados entre esses *softwares* usando três modelos simples de estruturas (por ordem crescente de complexidade: pilar, pórtico e estrutura porticada). Estes exemplos permitem fazer uma rápida análise de interoperabilidade dos programas em causa (ou até de outros programas visto que se trata de uma metodologia) nos domínios das informações analítica, geométrica e mecânica, que correspondem às áreas nas quais a presente dissertação se insere.

Esta análise é de grande importância já que a troca de informações entre *softwares* é cada vez mais uma necessidade no acto de engenharia e desenvolvimento de projecto.

Neste capitulo será feita uma análise com base nas ocorrências observadas, posteriormente o Capítulo 4 poderá contribuir com conhecimentos que permitam um melhor entendimento de algumas problemáticas na transferência de informações usando o modelo IFC, ao nível do seu funcionamento interno.

"We're starting to work with architects and engineering teams who can appreciate the value of interoperability. We'll produce case studies, measure metrics and be open about it. This industry is more driven by fear of losing a competitive advantage than being transformative or innovative."

- John Tocci, President, Tocci Building Companies

3.2. APLICAÇÃO À REALIDADE DE 2013

A realização destes testes de interoperabilidade tem como principal objectivo detectar os principais problemas decorrentes da passagem de informação entre os *softwares* Revit 2013, Robot 2013, Archicad 16 e Tricalc 7.4, sendo usado para esse efeito ficheiros de troca do esquema IFC2X3 e também se possível passagem directa de dados entre programas.

Neste estudo de interoperabilidade foi dada uma especial enfase aos programas "Revit" e "Robot" ambos da Autodesk. O programa Revit é um *software* vocacionado principalmente para a modelação

de edifícios mas possui também uma componente que permite a aplicação de cargas estruturais, atribuição de materiais e permite também já fazer algumas verificações ao nível das ligações estruturais. A criação de ficheiros é feita a partir de quatro tipos de *templates* que são: modelo base arquitectónico, estrutural, mecânico ou de construção.

Já o programa Robot é um *software* vocacionado essencialmente para a análise estrutural. As suas representações tridimensionais apenas procuram apresentar os elementos de forma simples e inteligível, não procuram permitir a renderização do modelo para apresentações formais do edifício. Ainda assim este modelo 3D gerado no Robot (ou transferido directamente do Revit ou indirectamente de outros programas usando a especificação IFC) permite a elaboração de cortes estruturais e outras peças de projecto de estruturas, constituindo uma mais-valia na aplicação prática destes softwares. O Revit também permite a elaboração dessas peças, mas uma vez que não permite fazer a sua verificação enquanto elemento de resistência estrutural considera-se que há uma maior vantagem em realizar-se esse tipo de trabalho directamente do Robot.

O Archicad da Graphisoft, já bem conhecido no mercado e assumido como sendo o primeiro programa BIM, é um *software* de modelação que permite sobretudo a criação de modelos arquitectónicos de edifícios, interiores e áreas urbanas e desenvolvimento de peças do projecto, competindo directamente com o Revit.

Quanto ao Tricalc difundido pela Arktec, é um programa de cálculo de estruturas metálicas, de betão, de madeira e de outros materiais, com possibilidade de escolha entre os Eurocódigos Estruturais ou os regulamentos nacionais como o RSA e o REBAP, ou até mesmo regulamentos de outros países como a Espanha, Brasil, México, Argentina e Estados Unidos da América. Umas das suas fortes vantagens são a possibilidade de pré-dimensionamento automático, cálculo automático de secções e peritagem detalhada. Existe uma afinidade especial entre o Tricalc e o Archicad, fomentada pela parceria entre a Arktec e a Graphisoft. Ao Tricalc pode ser associado o T-Connect para o cálculo de ligações de estruturas metálicas.

O esquema IFC2X3 foi o formato IFC que constituiu o principal elo que possibilitou a ligação e troca de informação entre todos estes programas, mas muitos outros poderiam ter sido usados. A buildingSMART disponibiliza a lista dos programas compatíveis com a especificação IFC. Como situação extra, foi também usado o esquema IFC4 enquanto formato que também permite a criação de modelos através da sua linguagem de código.

As conclusões retiradas da observação e experimentação prática são válidas para a realidade de 2013.

"If I was a software vendor I would make sure my technology was the most interoperable in the industry as my competitive advantage."

- Hector Camps, Educator and President PHI Cubed, Inc.

3.3. Definição da Estratégia de Avaliação

Para a simulação de testes de interoperabilidade foi usado um método de experimentação de troca de dados com suporte base no formato IFC, sendo os casos estudados sempre apenas entre dois *softwares* de cada vez onde um é identificado como sendo o programa origem e o outro o programa destino, havendo a possibilidade de inversão dos papeis. A transferência de dados de forma directa apenas foi possível nos casos de estudo entre o Revit e o Robot (e vice-versa) usando uma ferramenta incluída no

Revit que permite a exportação e importação directa de dados para o Robot. De referir que a comunicação directa de dados só é possível dentro de "famílias" de *software*, isto é, aplicações pertencentes a um mesmo distribuidor.

Qualquer caso de estudo deve identificar o âmbito a explorar, o propósito da sua exploração e os critérios pelos quais se fará a avaliação desse estudo (YIN 1993). O âmbito e o propósito do estudo já foram definidos anteriormente ficando em falta a definição dos critérios de avaliação. Para se poder fazer uma análise de transferência de informações definiu-se um conjunto de modelos com as principais variáveis a testar. Estas variáveis identificam as principais propriedades usadas num modelo estrutural que permitirão avaliar e classificar os níveis de interoperabilidade, isto é, será possível determinar quais os domínios de dados que são transferidos ou não. Os tipos de informação a introduzir no modelo de teste diferenciam-se pela aplicabilidade de cada ferramenta BIM, e por isso foram divididos de forma prática em variáveis de geometria, materiais, ligações, apoios, cargas e níveis. A variável "geometria" é um aspecto no domínio da arquitectura, tem caracter mais geral e é normalmente o aspecto mais saliente e mais facilmente visível. As variáveis de "materiais", "ligações", "apoios" e "cargas" enquadram-se no domínio da engenharia. A variável "níveis" é um aspecto com carácter misto. Estão então agora também definidos também os critérios de avaliação do estudo.

Na definição estratégica de análise representada na Figura 3.1 está subentendido o conceito de pesquisa sistemática que será baseada na descrição, classificação e quantificação dos resultados observados na execução dos testes de interoperabilidade, tentando-se encontrar possíveis causas para uma determinada falha e verificação da plausibilidade das conclusões retiradas.

Figura 3.1 – O formato IFC como método genérico para transferia de dados entre dois aplicativos BIM.

As variáveis já descritas constituem um agrupamento de informações, estas serão caracterizadas por particularidades que possibilitem identificar rapidamente se os dados na troca em questão manter-se-ão ou não inalterados na realização da transferência de um programa origem para um programa destino.

Suscintamente o diagrama da Figura 3.2 representa o processo usado para a realização de testes de interoperabilidade entre programas.

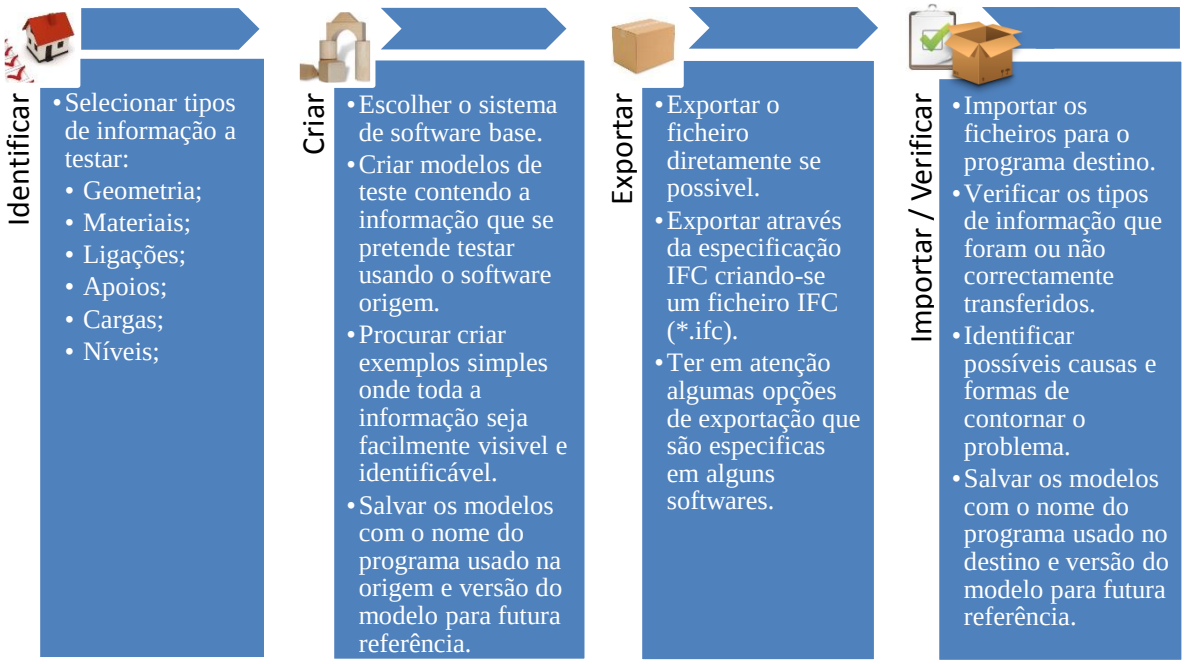


Figura 3.2 – Diagrama de procedimento para realização de testes de interoperabilidade

Portanto, este estudo empírico de interoperabilidade resulta no fundo da aplicação de um método de avaliação com um fluxo sequencial de testes entre dois programas, onde são avaliadas um conjunto de variáveis a testar. Este método permitiu também avaliar o que sucede no programa destino (BIM 2) quando alterada alguma informação no programa origem (BIM 1), retirando-se desse modo uma série de evidências que derivaram da análise do seguimento lógico. Na análise realizada no ponto 3.5 será apresentado um exame detalhado com os problemas detectados na transferência dos elementos.

Este tipo de análise deixa em aberto algumas hipóteses de resultado da transferência de dados. A situação desejável é uma transferência completa dos dados sem se verificar qualquer tipo de alteração na informação original. Porém, ainda é difícil de alcançar na situação actual. As situações mais verificadas são as situações onde o modelo gerado é transferido e aberto possibilitando a transferência das informações básicas como geometrias simples e algumas propriedades, evitando a remodelação total do edifício, todavia o nível de detalhamento é ainda bastante limitado, o que neste momento torna a especificação IFC de certa forma insuficiente. Infelizmente este não é o pior cenário, muitas vezes os modelos são transferidos e há uma grande perda de informação que obriga a reintrodução da maioria dos dados, contendo simplesmente informação geométrica de baixa complexidade (elementos rectos) e sem informações agregadas podendo ainda, na pior das hipóteses, não ser possível abrir o ficheiro de troca no programa destino ou ser aberto e não conter quaisquer dados.

3.4. Funcionamento Interno dos Softwares no domínio da Interoperabilidade

A conversão de um modelo a partir de um *software* para um ficheiro IFC faz-se pela identificação dos elementos simples que constituem esse modelo (dados geométricos, propriedades e relações). Depois pela discriminação de categorias numa tabela interna do programa, este faz a correspondência entre as suas categorias que suporta e a classe IFC respectiva que reproduz o mesmo tipo de informação. Devido à limitação que existe na quantidade de categorias da tabela de correspondência, muita informação é perdida logo à partida numa transferência de informação para o formato IFC. As

geometrias de baixa complexidade são um dos poucos tipos de dados que geralmente são transferidos sem que se relatem quaisquer tipos de problemas de interoperabilidade. Para o caso dos restantes tipos de informações como relações entre objectos, propriedades e até mesmo geometrias mais complexas (por exemplo elementos curvilíneos) são necessários determinados cuidados pois infelizmente será provável que ocorram problemas na passagem desse tipo de informações ditas mais complexas (Gequaltec 2011). A Figura 3.3 constitui um exemplo que retracta os problemas de interoperabilidade de que se falam. As geometrias simples das vigas lajes e pilares são transferidas sem nenhum problema visível enquanto as geometrias mais complexas do veículo geraram conflitos, originando uma visualização bastante distorcida e com perda de informação.

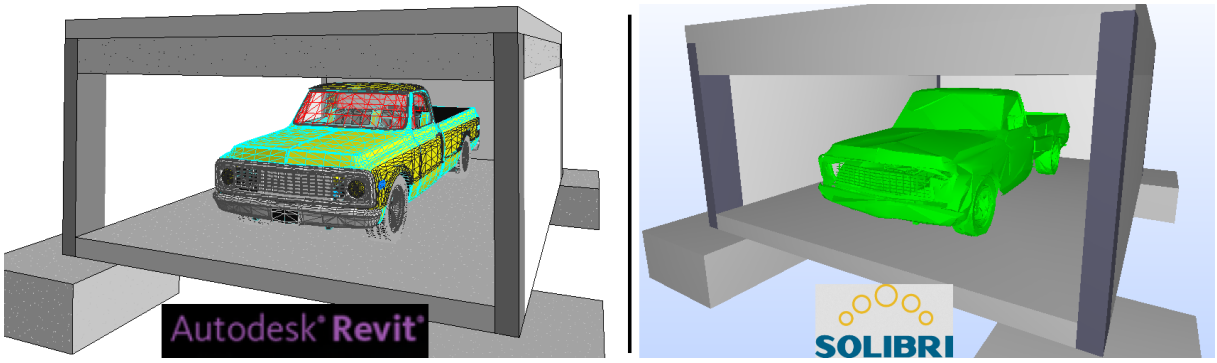


Figura 3.3 – Ficheiro IFC exportado pelo Revit e visualizado com o Solibri Model Viewer.

Dado que existe grande ocorrência deste tipo de problema, de modo a que fosse possível contorna-lo em parte, surgiram as propriedades conhecidas como "*Property Sets*". O modelo IFC faz a diferenciação entre atributos e propriedades. Os atributos estão directamente ligados ao objecto através da definição de um valor na secção de atributos de uma entidade e as propriedades são definidas individualmente formando-se depois um grupo de características e só depois então é que se faz a atribuição desse grupo a um determinado objecto através de um relacionamento entre entidades. Este tipo de propriedades tem evoluído ao longo do tempo e algumas delas podem agora ser específicas de um determinado projecto, região ou país. O esquema IFC faz a transição destas propriedades em conjuntos com nomes pré-definidos que identificam e agrupam as propriedades de um determinado tipo. O Archicad é um dos programas que recorre frequentemente ao uso das *Property Sets* para a exportação e importação de informação. Como se pode verificar pela Figura 3.4, há uma melhoria significativa na transferência de informação entre o Revit e o Archicad. Apesar de se continuarem a perder algumas informações relativas às diversas partes do veículo e materiais, observa-se que pelo menos toda a geometria permaneceu intacta.

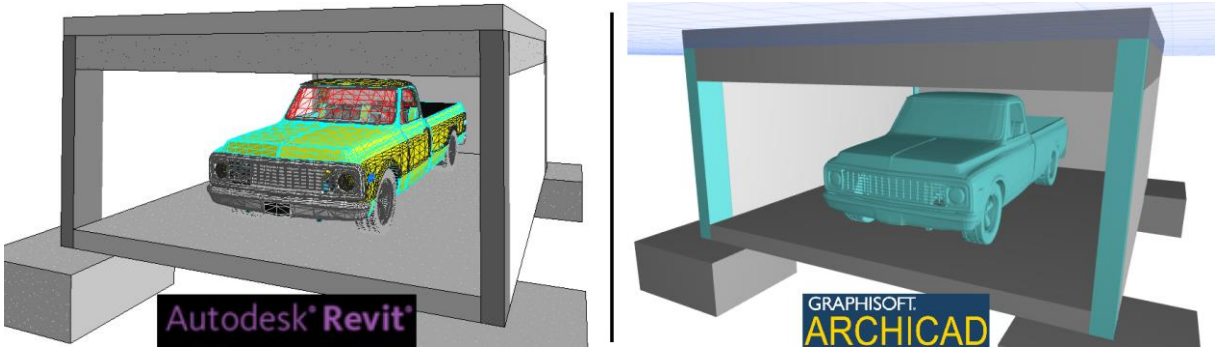


Figura 3.4 – Ficheiro IFC exportado pelo Revit e importado pelo Archicad.

3.4.1. REVIT

Para se aprofundarem os conhecimentos no Revit, devem ser feitas algumas ressalvas que são válidas para todos os casos que envolvam troca de informação com este programa. Na versão Revit 2013 as quatros vertentes (Arquitectura, Estruturas, Construção e Mecânica) fazem agora parte de um único programa do Revit, sendo organizadas por *templates* (modelos). Com o Revit surge o conceito de famílias que apresenta algumas semelhanças com o conceito de blocos das ferramentas CAD. As famílias são componentes usados na construção do modelo virtual como por exemplo os pilares, paredes, lajes, janelas, etc. Cada família poderá ser constituída por múltiplos tipos de informação com diferentes dimensões, materiais e outros parâmetros variáveis. As famílias podem ser criadas, editadas e importadas do exterior, cada alteração feita num dado tipo nesse modelo é aplicado a todos os elementos semelhantes de todo o projecto. Para importação de famílias que não existam por predefinição no Revit, entre outras pode ser usada a biblioteca de famílias da Autodesk. No caso dos seguintes exemplos que serão alvo de estudo foi necessário descarregar as famílias que representam as condições de fronteira e que não vêm previamente definidas no Revit e posteriormente introduzidas nas definições das condições de fronteira em cada um dos *templates* (Figura 3.5 e Figura 3.6).

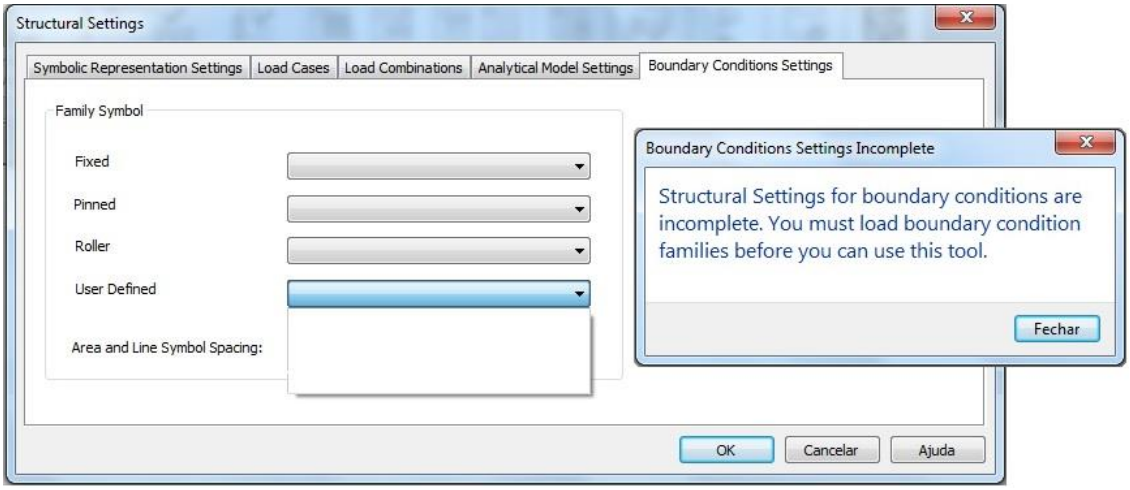


Figura 3.5 – Definições das condições de fronteira antes da introdução das famílias respectivas.

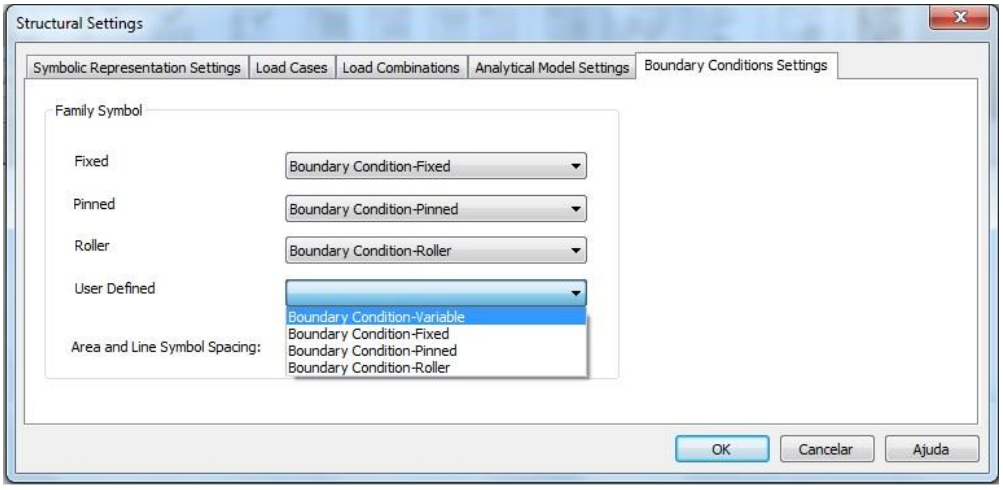


Figura 3.6 – Definições das condições de fronteira depois da introdução das famílias respectivas.

Conhecidas já as informações constantes nos passos de Identificação e Criação do procedimento de avaliação é necessário Exportar os modelos do programa origem para o programa destino. Tendo-se o Revit como programa origem são necessários alguns cuidados que poderão melhorar as condições de transferência (opção "*Send options"* da Figura 3.7).

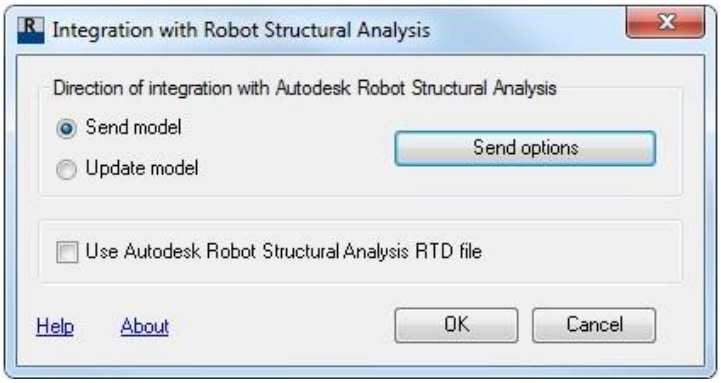


Figura 3.7 – Janela do Revit para a integração de modelos do Revit no Robot e vice-versa.

No caso de se tratar de uma transferência de forma directa do modelo do Revit para o Robot (opção "*Send model*") ou vice-versa (opção "*Update model*"), devem ser seleccionadas algumas opções de exportação, como a opção de ignorar o peso próprio para que desta forma se consiga melhor identificar se a informação relativa às cargas aplicadas à estrutura é transferida ou não para o Robot. Ao não ser ignorado o peso próprio da estrutura poderiam ser tomadas considerações erradas, uma vez que se fosse efectuado o cálculo da estrutura recém-chegada ao Robot seriam obtidos resultados, o que poderia fundamentar erradamente que a troca de informação relativa às cargas teria sido realizada correctamente quando na verdade a única acção que poderia ter sido transferida havia sido a acção do peso próprio. Ainda relativamente às opções de exportação da estrutura do Revit para o Robot deve haver o cuidado de se fazer algumas alterações nas configurações das opções básicas e avançadas. Deve ser seleccionada a opção "*Execute model correction in Robot*" para que sejam efectuadas as correcções necessárias para garantir uma melhor representação em termos estruturais, seleccionar também quanto às propriedades das barras a opção "*Do not use Revit settings*" pois as ligações do Revit são mais susceptíveis de erros estruturais do que as definições de ligações do Robot e ainda nas opções avançadas seleccionar "*Select material of the best matching parameters*" para que haja um maior rigor na troca de dados dos materiais como mostrado na Figura 3.8

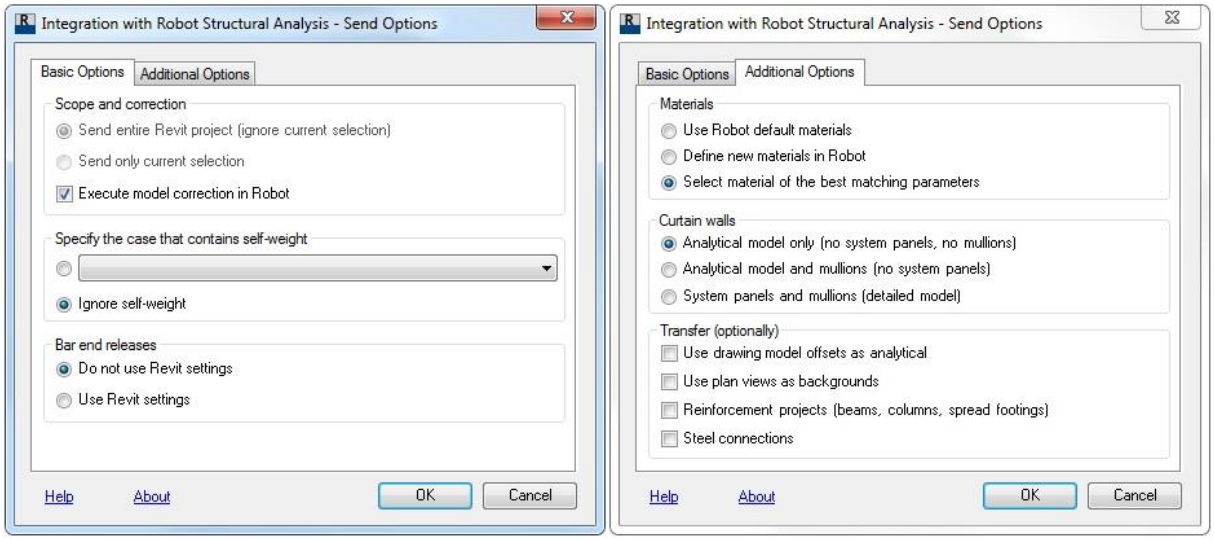


Figura 3.8 – Opções de exportação de modelos do Revit para o Robot de forma directa.

No caso de se tratar de uma transferência de dados usando ficheiros IFC importa referir a existência de uma tabela de relações entre as classes locais e as classes IFC para exportação e outra tabela para importação. Esse dicionário de correlações das categorias Revit com as classes IFC, para exportação de ficheiros, que vem instalado por defeito é bastante limitado pois estão em falta muitas categorias e dentro das categorias identificadas muitas não têm classe IFC correlacionada e por isso essa informação não poderá ser transferida para um ficheiro IFC. Este dicionário representado sob o formato de ficheiros "*.txt" com o nome "exportlayers-ifc-IAI.txt" não vem completo, no entanto é possível alterar e acrescentar novas correlações de classes e até existe a hipótese de importar um ficheiro mais completo que o substitua, contudo sem que haja qualquer tipo de instrução existe uma grande dificuldade em se determinar a correcta correlação de qual a classe IFC que corresponde a uma determinada categoria do Revit que não esteja especificada. A Figura 3.9 apresenta a referida tabela que faz a correspondência entre as categorias e as classes IFC. Como se pode apurar a maioria das categorias mencionadas nas opções do "IFC Export Classes" não têm classe IFC respectiva (atribuído "Not Exported"), para além de que as diferentes entidades IFC mencionadas em toda a tabela são apenas cerca de meia centena, quando existem já mais de 750 entidades IFC (IFC4). Dá-se conta da ausência de categorias fundamentais de definição analítica de estruturas e cargas estruturais como por exemplo o *IfcVertex*, *IfcMember*, *IfcStructuralLoad*, *IfcRelConnectsStructuralMember*, *IfcBoundaryNodeCondition*, *IfcSstructuralLoadCase* entre muitas outras, entidades sem as quais não é possível exportar dados relativos a nós, cargas, ligações entre elementos estruturais e condições de apoio. Posto isto, percebe-se que o Revit apenas exporta cerca de 7% das classes existentes, então, quanto mais complexo e detalhado for o projecto a exportar, maior e mais notável será a perda de dados quando efectuada a exportação para o IFC. É possível melhorar ligeiramente esta percentagem de eficiência na exportação, definindo manualmente as classes que correspondem às características atribuídas como "Not Exported" e completar assim a tabela, mas isso implica que haja o conhecimento de qual a classe correcta a atribuir para cada categoria.

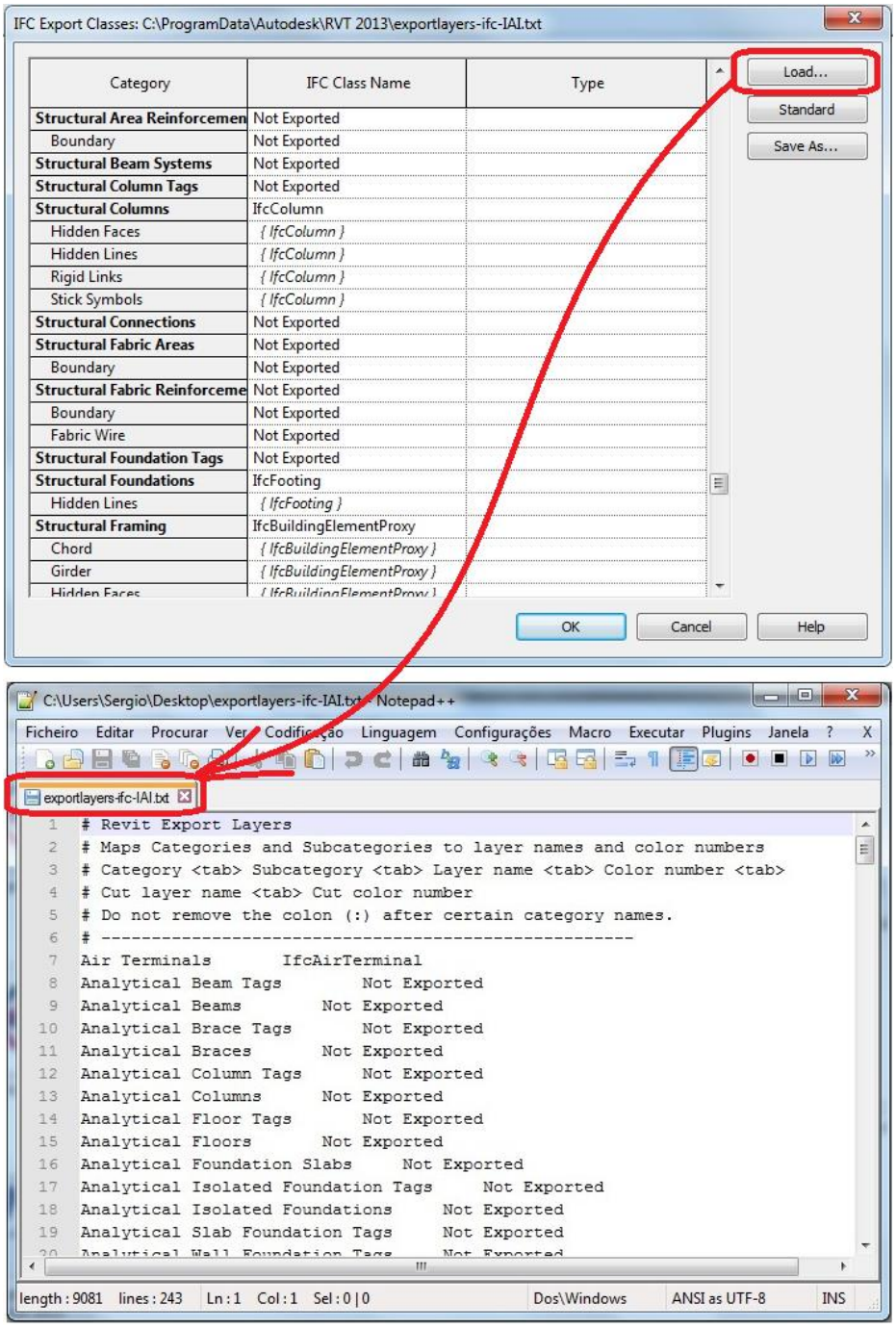
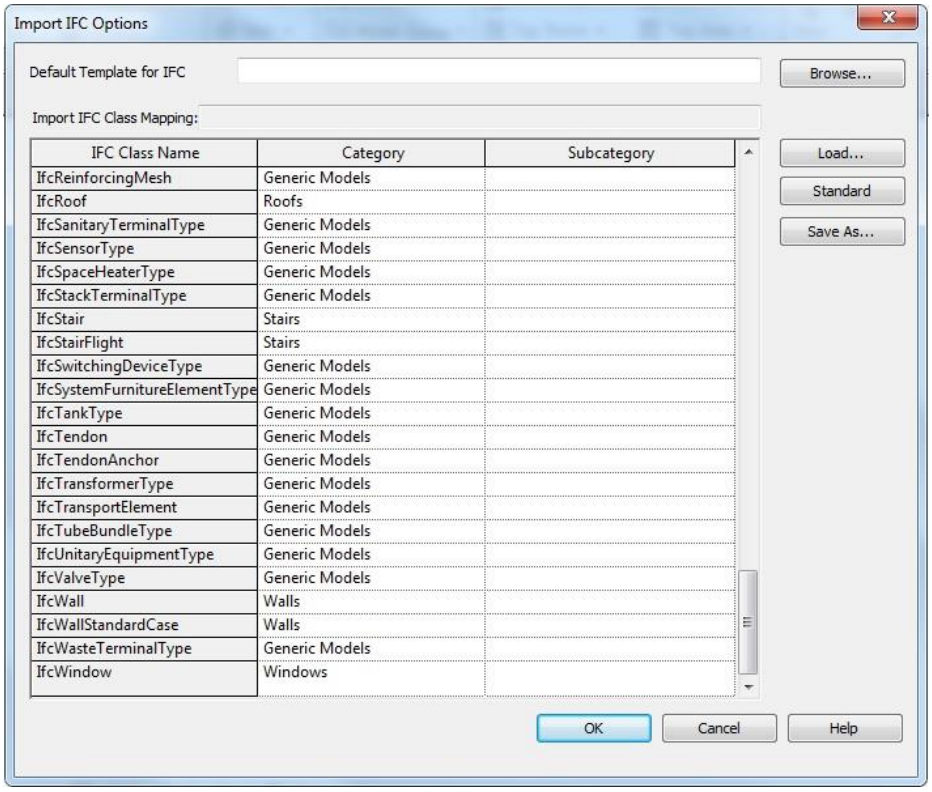


Figura 3.9 – Tabela de correspondência entre as categorias do Revit e as classes IFC sustentada pelo ficheiro exportlayers-ifc-IAI.txt.

A tabela de importação de ficheiros IFC para o Revit é semelhante à de exportação, mas desta feita temos como *input* a classe IFC e *output* a categoria do Revit. Mais uma vez a lista apresenta-se bastante incompleta, com pouco mais de uma centena de entidades IFC identificadas para importação, comportando ainda assim apenas cerca de 14% das entidades existentes e novamente a sua maioria das classes IFC não tem categoria especifica correspondente (atribuído "Generic Models"). A origem da informação contida na tabela de importação é feita nos mesmos moldes que na tabela de exportação

para IFC, ou seja, também é feita a partir de um ficheiro de texto "*.txt" mas desta vez denominado "exportlayers-dwg-ISO13567.txt". Pelo processo esse já explicado anteriormente poderão ser especificadas categorias para as classes IFC identificadas, mas desta feita existe uma dificuldade ainda maior em determinar as categorias existentes do Revit que correspondam às classes IFC, conforme exposto na Figura 3.10.



IFC Class Name	Category	Subcategory
IfcReinforcingMesh	Generic Models	
IfcRoof	Roofs	
IfcSanitaryTerminalType	Generic Models	
IfcSensorType	Generic Models	
IfcSpaceHeaterType	Generic Models	
IfcStackTerminalType	Generic Models	
IfcStair	Stairs	
IfcStairFlight	Stairs	
IfcSwitchingDeviceType	Generic Models	
IfcSystemFurnitureElementType	Generic Models	
IfcTankType	Generic Models	
IfcTendon	Generic Models	
IfcTendonAnchor	Generic Models	
IfcTransformerType	Generic Models	
IfcTransportElement	Generic Models	
IfcTubeBundleType	Generic Models	
IfcUnitaryEquipmentType	Generic Models	
IfcValveType	Generic Models	
IfcWall	Walls	
IfcWallStandardCase	Walls	
IfcWasteTerminalType	Generic Models	
IfcWindow	Windows	

Figura 3.10 – Tabela de correspondência entre as classes IFC e as categorias do Revit e sustentada pelo ficheiro exportlayers-dwg-ISO13567.txt.

3.4.2. ROBOT

O programa Robot não permite a exportação de ficheiros para o formato IFC o que limita desde logo muitas potencialidades deste *software*, contudo permite uma importação mas também ela limitada. Como alternativa para exportação no Robot tem-se o formato CIS/2 alternativo ao IFC mas válido apenas para estruturas metálicas.

A transferência de informações provenientes do Robot usando a ferramenta de integração incluída no Revit acima descrita faz-se de forma semelhante mas desta vez seleccionando-se a opção "Update model" da janela mostrada na Figura 3.11. Nas opções de envio é possível seleccionar se a transferência se realizará a nível de todo o modelo ou apenas a nível da parte estrutural. Opcionalmente ainda é permitido importar os resultados como reacções e forças internas, ligações metálicas e reforços necessários em elementos o que por exemplo permite a substituição de um elemento que não verificou as condições da segurança no Robot e cuja secção será alterada de modo a que se verifiquem essas condições.

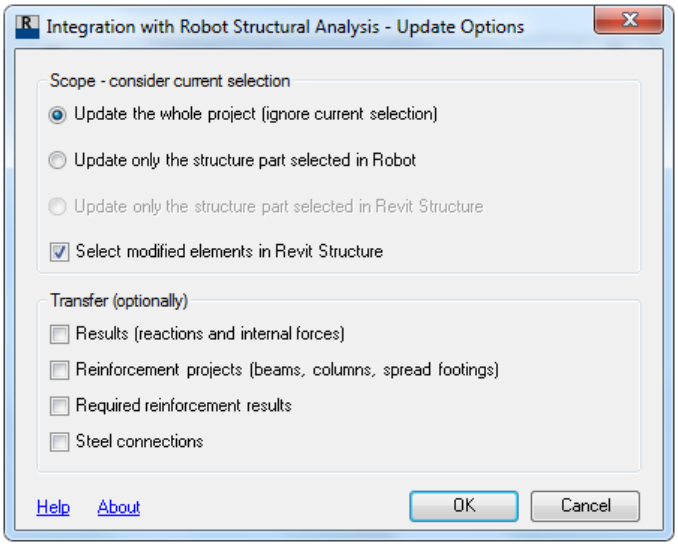


Figura 3.11 – Opções de importação de modelos do Robot para o Revit de forma directa.

3.4.3. ARCHICAD

O Archicad não possibilita a troca directa de informação com nenhum programa, o seu funcionamento em termos de transferência de dados baseia-se apenas no modelo IFC e opera de forma semelhante ao observado no Revit. Existe uma lista (Figura 3.12), também ela limitada, de classes IFC que são susceptíveis de ser exportadas, mas aqui não é mostrada ao utilizador a relação interna que há entre as categorias do Archicad e as classes do IFC.

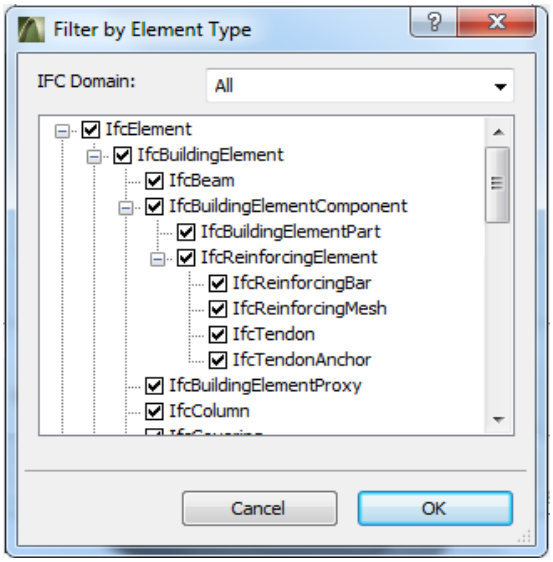


Figura 3.12 – Lista de classes IFC exportadas pelo Archicad

Em termos de transferência de dados para o modelo IFC, uma das grandes vantagens do Archicad em relação ao Revit é oferecer a possibilidade de manuseamento das *Property Sets* especificas para cada elemento (Figura 3.13). Desta forma é possível assegurar que determinada propriedade específica que se pretende atribuir a um dado elemento é transferida aquando da criação do ficheiro de troca IFC, controlando-se de certa forma a informação que é passada para esse ficheiro. A Figura 3.13 mostra um exemplo de *PropertySets* aplicáveis a um pilar

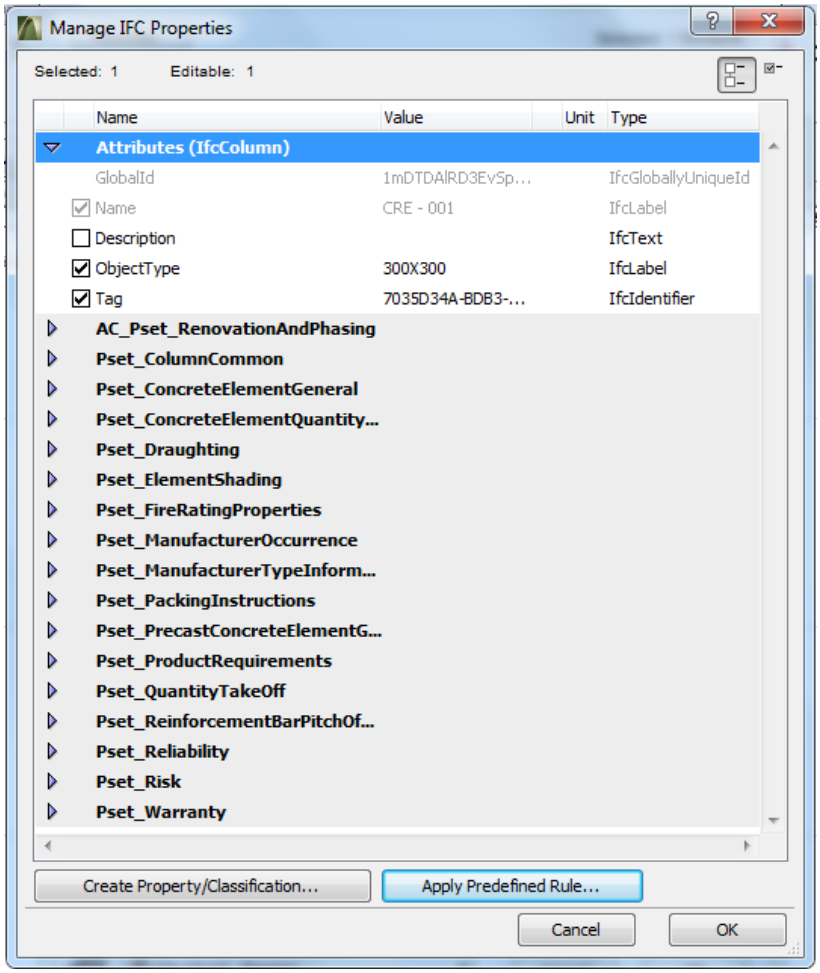


Figura 3.13 – Exemplo de *PropertySets* aplicáveis no Archicad.

3.4.4. TRICALC

No Tricalc não são disponibilizados quaisquer quadros informativos identificando as classes IFC que são admitidas para exportação e importação, por isso, esta questão poderá apenas ser avaliada aplicando o procedimento para realização de testes de interoperabilidade definido no ponto 3.3 que permitirá registar os problemas ocorridos na transferência dos modelos criados.

3.4.5. NIST IFC FILE ANALYSER

O programa "IFC File Analyser" do NIST constituiu um auxílio na análise dos ficheiros IFC gerados em todos os processos de troca via IFC. Revelou ser uma utilidade que constitui uma mais-valia na análise dos ficheiros exportados, permitindo contabilizar o número de entidades IFC exportadas e mais do que isso permitindo saber quais os tipos de entidades que foram exportados, suas relações e principais atributos através da criação de um ficheiro do tipo "*.xlsx" agrupando os tipos de informação. Dessa forma, perante a observação de um problema é possível confirmar rapidamente se a origem desse problema se devia ou não à ausência de uma entidade.

3.5. TESTES DE INTEROPERABILIDADE

Dando-se início à execução prática de testes de interoperabilidade, com um estudo num domínio mais voltado para a definição estrutural, serão elaborados alguns casos de estudo envolvendo os programas descritos e testando os três modelos simples já mencionados. Foi então modelado um pilar [1], um pórtico [2] e uma estrutura porticada [3] para avaliar o fluxo de processos e respectivo mapeamento que possibilita determinar as capacidades e níveis de interoperabilidade. Procurou-se definir um material semelhante para todas os elementos estruturais para se verificar se é mantida a coerência de propriedades, neste caso os materiais foram definidos como sendo de betão. Em anexo apresentam-se representações do pilar nos diversos casos para dar uma ideia visual das diferentes situações.

1) A geometria do pilar criado caracteriza-se por uma secção rectangular 0.20x0.30m e 3 metros de altura. Sempre que possível serão aplicadas duas cargas, uma pontual central à secção no topo do pilar (F1= -2,54 kN) e outra linear transversalmente ao pilar ao longo de toda a sua extensão (F2= -1,64 kN/m). Houve o cuidado de atribuir uma determinada peculiaridade aos valores das forças para que estas sejam mais fáceis de procurar e identificar em futuras trocas de dados, podendo-se assim rapidamente verificar se essa informação foi ou não transferida correctamente. As forças foram também definidas como sendo diferentes casos de carga, a força F1 diz respeito a uma carga permanente enquanto a força F2 diz respeito a uma acção do vento. Por fim, no domínio das acções, ter-se-á em conta um caso de carga para o estado limite último, denominado "Caso de Carga 1", do tipo "Combinação", em que a fórmula de cálculo será "1,35 x F1 + 1,50x F2". As características do betão a atribuir aos elementos serão aplicadas escolhendo um material pré-definido do programa de origem que estiver em causa.

2) O pórtico tem como função adicional verificar se a ligação dos elementos pilares e viga continua a verificar-se depois do modelo ser transferido do *software* BIM 1 para o *software* BIM 2. É constituído por dois pilares de igual secção (0.20x0.30m) cujo os eixos distam 5.5 metros entre si e uma viga com uma secção 0.20x0.40m que une o topo dos dois pilares. Foi também definida uma carga trapezoidal transversal à viga e com sentido vertical descendente, com -5 kN no ponto inicial e -2 kN no ponto final da viga.

3) Por último a estrutura porticada introduz como informação adicional aos testes de interoperabilidade anteriores uma laje com 0.22m de espessura e uma carga uniformemente distribuída nessa laje com sentido vertical descendente e um valor de -4 kN/m². Este modelo tem como principal função verificar a ligação entre elementos na troca de dados entre programas, especialmente as ligações entre a laje (elementos planos) e os pilares e vigas (elementos lineares) que correspondem às ligações mas susceptíveis da ocorrência de erros.

3.5.1. CASO A – REVIT ARCHITECTURE PARA ROBOT

3.5.1.1. Via Directa

Conforme pretendido modelou-se as três estruturas respeitando as secções e dimensões pretendidas, atribuindo a todas um material pré-definido do Revit, denominado "Concrete, Cast-in-Place, Gray" [Propriedades Térmicas (α=1x10^{-6}/°C); Propriedades Mecânicas (E=29250.0 MPa; ν=0.17; G=9964.0 Mpa; ρ=2407.31 kg/m³; ζ=0.15); Propriedades Betão (Compressão=24.1 MPa; Shear Strength Modification=1.0; Yield Strength=2.4 MPa; Tensile Strength=2.4 MPa)].

Usando o modelo de arquitectura do Revit é revelada uma incapacidade em adicionar condições de fronteira, já que os nós não são gerados neste *template*. O próprio Revit faz uma chamada de atenção para o facto de que os elementos não se encontram suportados (Figura 3.14).

Figura 3.14 – Janela de aviso apresentada pelo Revit Architecture.

Existe a possibilidade de criar cargas e combinações de cargas, mas estas só estarão relacionadas com a estrutura se aplicadas no eixo baricêntrico da secção rectangular. Esta afirmação resulta do caso testado experimentalmente, apenas a força F1 (DL1) aparece representada depois de se fazer a transferência para o Robot, a força F2 (WIND1) perde-se no processo de troca, contudo, é mencionada na combinação que havia sido definida. No teste com o pórtico [2] aplicou-se a carga vertical descendente ao eixo baricêntrico da viga, e como esperado a carga foi transferida sem problemas a relatar.

Para se comprovar melhor este facto, foi colocada uma terceira força pontual, F3, centrada no topo do pilar [1], no mesmo ponto de aplicação da força F1. Ao se exportar o modelo para o Robot as forças F1 e F3 são identificadas se aplicadas no centro da peça, mas se atribuirmos uma excentricidade à força F3 em relação ao eixo baricêntrico da secção rectangular, esta já não é identificada pelo Robot, sendo contabilizada apenas para efeitos de cálculo de combinações (já definidas no Revit). Por razões do domínio da programação que não fazem parte do âmbito da Engenharia Civil, não é possível apurar as causas.

Quanto a conexão entre elementos, para se garantir uma correcta passagem da informação por via directa os elementos devem ser unidos nos pontos centrais, por exemplo, uma viga deve unir o ponto central de dois pilares. Caso não se proceda desta forma ao transferir o modelo para o Robot os elementos não aparecerão unidos num mesmo nó, situação apresentada na Figura 3.15.

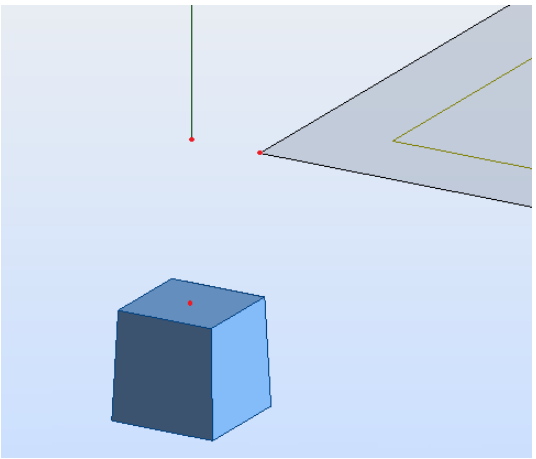


Figura 3.15 – Desconexão entre apoio, pilar e laje.

3.5.1.2. Via IFC

Posteriormente as estruturas foram exportadas para ficheiros IFC, originando-se ficheiros com extensão"*.ifc" que também podem ser abertos com a extensão "*.txt", apenas para se ter uma noção da dimensão do ficheiro de texto criado, a definição do pilar [1] pelo Revit gerou 373 instâncias IFC. Ao abrir o ficheiro IFC do pilar no Robot, denota-se que o Robot identifica o pilar e representa-o genericamente através de um eixo onde define nós em ambas as extremidades do elemento, havendo também a possibilidade de visualizar a secção e geometria do pilar. A informação do membro estrutural foi acrescentada pelo Robot, já que não há indicação específica de nós nem no ficheiro inicial criado no Revit, nem no ficheiro IFC em questão.

Quanto às restantes informações pode dizer-se que apenas a informação geométrica permanece intacta, as restantes informações são suprimidas ou alteradas. Deixam de existir quaisquer informações acerca de cargas e grelhas e apenas é identificado o nível onde o pilar está aplicado.

Quanto à informação relativa ao material é um ponto que merece estudo mais aprofundado, pois no ficheiro IFC exportado pelo Revit, aparece mencionado no *IfcMaterial* o material "Concrete – Cast-in-Place Concrete" mas abrindo esse ficheiro no Robot o material atribuído é "BETÃO". Observou-se também que não são definidas nenhumas propriedades físicas das já referidas que faziam parte das características do material. Para se conseguir averiguar qual a fonte que assegura a informação dos materiais dos elementos ao Robot, investigaram-se as classes que estão directamente relacionadas com materiais existentes no ficheiro IFC criado, como é o caso do *IfcMaterial*, *IfcRelAssociatesMaterial* e *IfcMaterialDefinitionRepresentation*, excluíram-se essas classes e voltou-se a abrir o ficheiro IFC modificado. Novamente o material que está definido para o pilar é o material "BETÃO". Eliminando as classes onde é mencionada a palavra "concreto", como é o caso das classes *IfcCurveStyle*, *IfcCurveFont*, *IfcCurveFontPattern* e IfcSurfaceStyle também não altera os dados do material e voltando a abrir no Robot o ficheiro IFC já modificado pela segunda vez, volta-se a verificar que o material associado ao pilar é novamente "BETÃO". Por fim, eliminaram-se todas as classes, excepto as estritamente necessárias para gerar a representação de um pilar e mais uma vez o material obtido ao abrir o ficheiro IFC no Robot foi "BETÃO". Apenas para que todas as dúvidas fossem dissipadas, definiu-se o pilar como sendo de madeira, exportou-se para IFC e mais uma vez abriu-se o ficheiro IFC no Robot, e como já seria de esperar o material definido não era madeira mas sim "BETÃO". Assim sendo, podemos concluir que o ficheiro IFC gerado pelo Revit, não contém informação acerca das propriedades físicas dos materiais e que o Robot atribui aos elementos o material "BETÃO", por predefinição.

3.5.2. CASO B – REVIT STRUCTURES PARA ROBOT

3.5.2.1. Via Directa

Modelando a partir do modelo estrutural do Revit, foram seguidos os mesmo passos que no "Caso A" e atribuídas as mesmas características aos elementos estruturais. Destaca-se que agora já é possível adicionar condições de fronteira e portanto, foram adicionados encastramentos nas bases dos pilares das três situações. Verifica-se que ao transferir o modelo para o Robot a informação acerca dos apoios é correctamente transferida, assim sendo, ao contrário do ocorrido no "Caso A", desta feita a estrutura não terá problemas de suporte (Figura 3.16).

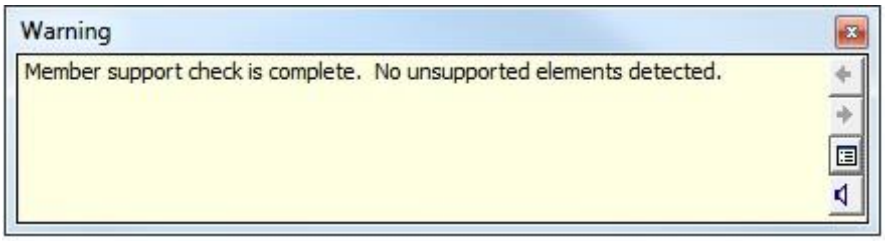


Figura 3.16 – Janela de verificação apresentada pelo Revit Architecture.

Outra das vantagens do modelo Revit Structures é a identificação dos elementos também através da sua definição analítica, o que por exemplo facilita na visualização das conexões entre elementos. Desta forma garante-se uma melhor funcionalidade das ligações entre os diferentes tipos de elementos, sendo até possível atribuir uma certa excentricidade em relação ao eixo baricêntrico mantendo-se a ligação.

Como se pode verificar pela Figura 3.17, onde à esquerda apresenta-se a ligação com as vigas unidas directamente ao centro do pilar e na ligação à direita uma das vigas apresenta uma pequena excentricidade nessa ligação mas a sua definição analítica permanece inalterada.

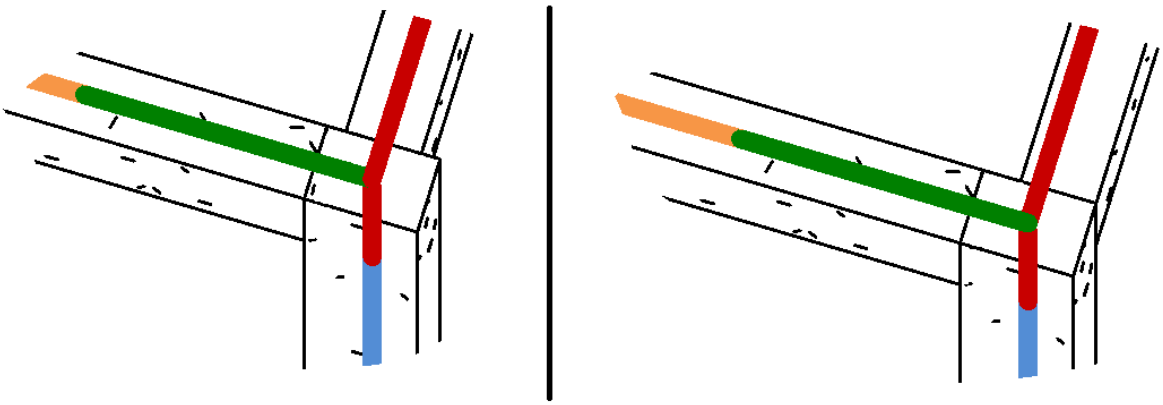


Figura 3.17 – Ligação entre elementos.

Outra situação recorrente que gera alguns problemas na ligação entre elementos é uma situação que se origina na criação de uma viga que faz o alinhamento entre mais do que dois pilares. Se para a criação dessa viga apenas for indicado o pilar que corresponde ao apoio inicial da viga e o pilar que corresponde ao apoio final da viga, poderão não ser estabelecidas ligações entre os pilares intermédios e a referida viga. Este problema pode resultar de questões relacionadas com a precisão do modelo. Pequenas imprecisões poderão implicar que as barras dos diferentes elementos não se intersectam e assim não são produzidas as ligações. A título de exemplo, se tivermos 3 pilares implantados em três pontos diferentes, o primeiro implantado no ponto A (0;0), o segundo implantado no ponto B (0;4,9999999) e o terceiro implantado no ponto C (0;10) e for definida uma viga pela indicação do seu ponto inicial (A) e ponto final (C), então o ponto B não ficará conectado com a viga porque há uma ligeira imprecisão que o coloca fora do alinhamento, apesar de formalmente o programa poder até indicar que o as coordenadas do ponto B são (0;5,0) por uma questão de arredondamentos, induzindo o utilizador em erro. Por essa razão é recomendada a criação de vigas através da sucessiva “piquetagem” dos pilares onde esta apoia.

A criação de *Rigid Links* (Figura 3.18) são outra hipótese que soluciona este tipo de problemas. Um *Rigid Link* (ou ligação rígida) é uma propriedade usada para estabelecer ligações analíticas entre pilares e vigas, restringindo o movimento relativo entre eles. Em aplicações de análise é definido como sendo um elemento infinitamente rígido sem peso. Pelo recurso ao uso de elementos rígidos soluciona-se este tipo de problemas com relativa facilidade, mas requer que o utilizador esteja sensível para estas questões. Quanto às restantes informações repete-se o sucedido no "Caso A".

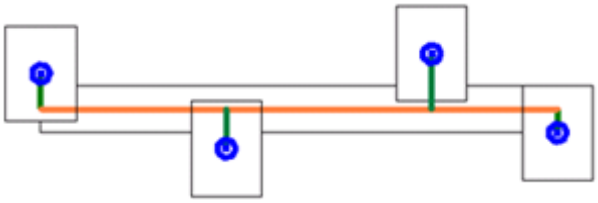


Figura 3.18 – Exemplo de pilares unidos estruturalmente ao eixo analítico da viga através de *Rigid Links* (representados a verde). (Autodesk 2013)

3.5.2.2. Via IFC

O processo realizado foi idêntico ao descrito no "Caso A" via IFC, com a diferença de que desta vez o ficheiro IFC gerado contém 373 classes. Contudo nesta versão foram suprimidas as classes *IfcAnnotation* e o *IfcDraughtingPreDefinedCurveFont*. Apesar disso a informação transferida e a informação perdida foi a mesma que a no "Caso A" via IFC.

3.5.3. Caso C – Robot para Revit Architecture

3.5.3.1. Via Directa

Uma vez que foram detectados alguns problemas de interoperabilidade na troca de dados criados no Revit e enviados para o Robot, decidiu-se fazer o processo inverso para se averiguar se desse modo existe ou não perda de informação. Ao se fazer a troca da informação do Robot para o Revit Architecture, verifica-se que deixam de existir cargas aplicadas à estrutura, bem como as ligações e os apoios. Contudo, os dados geométricos e dos materiais permanecem intactos. Conclui-se portanto que a possibilidade de definição de cargas no Revit serve apenas para envio para o Robot ou até mesmo exclusivamente para representação de forças no Revit ou anotação de informações adicionais mas sem vista a um cálculo no Revit.

3.5.3.2. Via IFC

Verificou-se que através do Robot não é possível exportar ficheiros para o IFC, contudo é possível importar. De um ponto de vista pessoal, esta situação leva a presumir que a Autodesk pretende que a informação gerada pelo Robot seja de acessibilidade reduzida por outros *softwares*. Talvez pela grande presença e influência que a Autodesk exerce no mercado actual, esta considere que não há necessidade de assegurar uma boa comunicação com programas concorrentes, optando assim por atribuir aos seus programas um elevado nível de inacessibilidade tornando-os bastante isolados ou de comunicação bastante limitada.

3.5.4. Caso D – Robot para Revit Structures

3.5.4.1. Via Directa

Este caso é bastante semelhante ao caso anterior, mas apresenta-se como uma variante. Desta feita a troca de informação foi efectuada entre o Robot e o Revit Structures. Curiosamente verificou-se que ao contrário do que se passou no Revit Architecture, aqui perdeu-se a informação relativa ao material. Uma possível justificação plausível para a situação constatada pode residir no facto de que o modelo de arquitectura do Revit faz a definição do material através de uma indicação criada pelo programa origem do tipo *Ifclabel* fazendo corresponder um material do mesmo nome, se existente na sua biblioteca. O modelo de estruturas do Revit não se contenta apenas com a informação do nome do material, este procura uma definição mais profunda das características físicas do material visto ter um caracter mais voltado para o cálculo estrutural, não encontrando nenhum desse tipo de informação não atribui ou cria qualquer material.

Por outro lado, foram transferidos dados correctamente que não tinham sido possíveis de transferir no Revit Architecture, como foi o caso das ligações e dos apoios, que neste caso apareceram correctamente definidos tal como no modelo original do Robot. Quanto às cargas, casos de carga e combinações de carga, mais uma vez essa informação foi perdida no processo de troca.

3.5.4.2. Via IFC

As conclusões retiradas para o “Caso C” via IFC são também aplicáveis para o “Caso D” via IFC.

3.5.5. Caso E – Revit Structures para Revit Structures/Archicad

A melhor forma de se detectar quais os dados que se perdem ao exportar e importar ficheiros IFC para o Revit é criar uma determinada estrutura, exportá-la para o IFC e abrir esse ficheiro IFC exportado novamente com o Revit. Deste modo é possível avaliar a quantidade de informação que se perde no processo. Ao executar este procedimento constata-se que apenas a informação geométrica é salvaguardada, toda a restante informação deixa de existir ou fica bastante incompleta, como é o caso dos níveis, no ficheiro inicial existiam vários níveis enquanto agora apenas existe o nível onde as estruturas estão implantadas.

A informação referente ao material dos elementos é mais uma vez aquela que requer mais atenção. O material aparece definido e pode-se verificar que se trata do material originalmente atribuído, mas apenas porque o Revit identifica o nome sugerido na classe *IfcSurfaceStyle* e procura um material na sua base de dados que tenha o mesmo nome. Caso encontre um material com o mesmo nome atribui ao pilar as características desse material, caso não encontre cria um novo material com o nome definido nessa classe IFC, mas sem características e propriedades físicas definidas. Posto isto é fácil compreender os problemas que esta situação pode comportar, percebe-se que o ficheiro IFC não contém em si informação sobre as características físicas dos materiais, apenas a indicação de qual o material que deve ser usado pelo programa destino, pelo que deve-se assegurar que o material mencionado no ficheiro IFC original consta na base de dados de materiais do programa no qual esse ficheiro irá ser executado e que esse material corresponde ao pretendido.

Para se tentar perceber melhor o mecanismo de interoperabilidade entre o IFC e o Revit no âmbito da definição dos materiais, eliminou-se a classe *IfcSurfaceStyle* para verificar se deixaria de existir um material associado aos elementos. Observa-se que na ausência desta classe o Revit vai buscar a

informação do material ao *IfcMaterial*, conclusão essa que se pode tirar dando diferentes nomes ao material na classe *IfcSurfaceStyle* e *IfcMaterial*, depois constatando o nome escolhido pelo Revit conclui-se que a classe que dá a primeira indicação de qual o material a definir pelo Revit é a classe IfcSurfaceStyle. Na ausência desta classe a informação é dada pelo *IfcMaterial*. Eliminando também o *IfcMaterial* consta-se que nenhum material é definido para os elementos.

Quanto à transferência de dados entre o Revit Structures e o Archicad, as ocorrências verificadas foram semelhantes ao sucedido na transferência de dados do Revit Structures novamente para o Revit Structures.

3.5.6. Caso F – Archicad para Tricalc

Usando-se o Archicad como programa de origem da informação denotam-se algumas limitações na introdução de dados, não é possível introduzir condições de apoio, nem aplicar cargas às estruturas, logo à partida não poderão ser classificados os níveis de interoperabilidade para estes tipos de dados. As ligações entre vigas e pilares são asseguradas pela representação de uma linha que serve de guia na verificação da correcta posição da ligação pretendida.

Enquanto programa de origem o Archicad tem o benefício de facilitar a inclusão de informações pela atribuição de *Property Sets* aos elementos, o que só por si já revela uma preocupação acrescida em relação aos outros *softwares*, de qualquer forma continua a ser insuficiente e nem sempre possibilita comunicar as informações pretendidas.

Uma das situações curiosas do Archicad é a possibilidade oferecida de exportação do modelo IFC consoante o programa destino (Figura 3.19). Visto que o IFC é um formato aberto que não estabelece nenhuma relação de afinidade específica com nenhum programa, não se compreende o porquê da diferenciação proposta pelos diferentes tradutores. Nos testes efectuados foi usado o tradutor genérico, seria necessário um estudo mais aprofundado para se averiguar quais as diferenças entre os diversos tradutores.

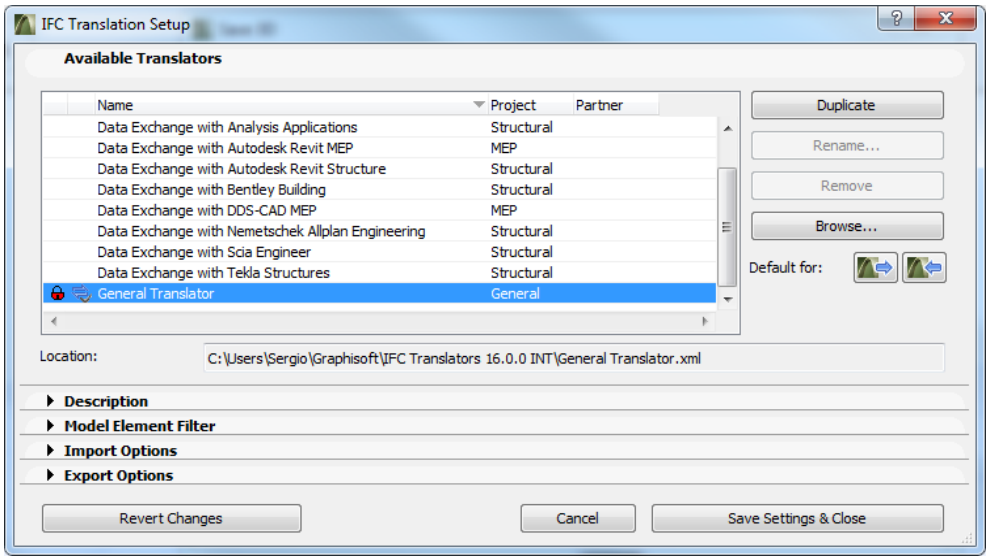


Figura 3.19 – Opções de tradução do modelo IFC usando o Archicad.

Depois de exportados os modelos pelo Archicad, ao se abrir esses modelos com o Tricalc surgem opções de importação que admitem a escolha de apenas algumas entidades IFC para facilitar a análise da estrutura eliminando-se a informação que não é necessária para esta etapa. Como se pretende

estudar os níveis de interoperabilidade não se excluiu nenhuma entidade. As trocas efectuadas concluíram que as informações geométricas e ligações foram bem transferidas, os restantes dados perderam-se no processo de troca.

3.5.7. Caso G – Tricalc para Archicad/Revit/Robot

Nos testes realizados onde o Tricalc é o programa de origem, verificou-se que na importação tanto com o Archicad, com o Revit ou com o Robot, apenas a informação geométrica é mantida intacta, todas as outras informações são transferidas parcialmente ou modificadas. Os materiais mantêm o nome na importação com o Archicad, mas são desprovidos de características físicas.

3.5.8. Caso H – Tricalc para Tricalc

Neste caso, para além dos três casos genéricos estudados, foram elaboradas duas estruturas, uma criada recorrendo à ferramenta "Rede" e outra recorrendo à ferramenta "Nave", ambas do separador "Geometria" no Tricalc. Depois de exportadas para o IFC, verifica-se que em termos de código e linguagem IFC a estrutura criada a partir da ferramenta "Rede" é bastante semelhante à estrutura criada usando a ferramenta "Nave", mas esta última contempla a classe "*IfcMember*". O ficheiro da estrutura criada usando a ferramenta "Rede" não comtempla essa classe, mas ainda assim ao voltar-se a abrir esse ficheiro no Tricalc, os elementos aparecem definidos e representados por um eixo que define o membro estrutural desses elementos. Esta ocorrência conduz à conclusão de que esses membros são gerados automaticamente pelo Tricalc quando este identifica elementos estruturais, supõe-se que o Tricalc filtre essa informação ao identificar classes que definem a função dos elementos, como por exemplo o *IfcColumn*, *IfcBeam*, *IfcSlab*, entre outras.

A troca de informações com estas estruturas ligeiramente mais complexas que já criam algumas situações onde os elementos se desconectam. O Tricalc tem uma ferramenta que permite corrigir este tipo de situações (Figura 3.20), mas não é possível afirmar que possibilita a resolução de todos os problemas de ligações porque a complexidade não era ainda muito elevada.

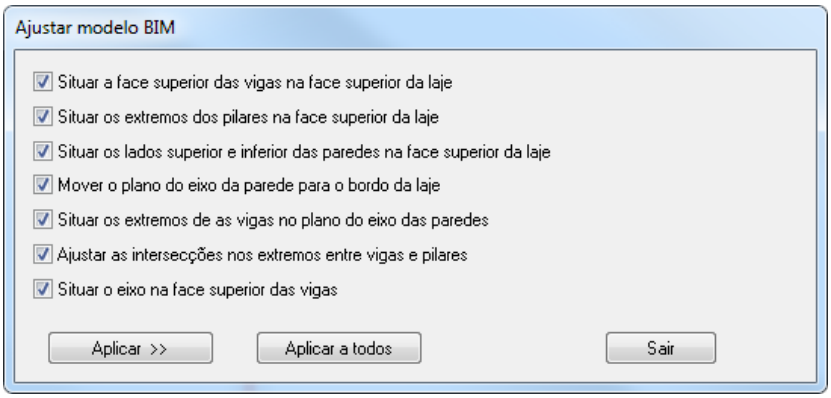


Figura 3.20 – Ferramenta de ajuste de nós dos modelos importados para o Tricalc via IFC.

3.5.9. Caso I – IFC4 para Revit/Robot/Archicad/Tricalc

Neste caso as estruturas modeladas foram criadas directamente a partir de um modelo de texto com base no esquema IFC4. Nenhum dos programas permitiu a visualização das estruturas criadas com este esquema dado ser um esquema ainda bastante recente. Apenas foi possível visualizar o modelo usando o programa Constructivity. No Capitulo 4 será exposta a construção do modelo da estrutura porticada com algumas alterações.

3.6. QUADRO RESUMO

Para concluir apresenta-se o quadro resumo de interoperabilidade (Quadro 1), que faz a descrição sucinta das principais ocorrências. Note-se que a geometria dos modelos foi sempre correctamente transferida. A correcta transferência de dados geométricos por si só trás já grandes benefícios, suprimindo a necessidade de remodelação da estrutura. Tratando-se apenas de uma parte da informação que um modelo BIM pode conter, consegue-se imaginar o potencial que os modelos teriam se fosse já possível transferir correctamente entre programas todas as informações pretendidas.

Quadro 1 – Interoperabilidade entre diferentes *softwares*

Casos	Origem	Destino	Informação na Origem	Informação no Destino			
				Via Directa		Via IFC2X3	
A	Revit Architecture (imagem)	Robot	Geometria☑ Materiais☑ Ligações☒ Apoios☒ Cargas☑ Níveis☑	Geometria Materiais Ligações Apoios Cargas Níveis	T T S S P T	Geometria Materiais Ligações Apoios Cargas Níveis	T P S S N P
B	Revit Structures	Robot	Geometria☑ Materiais ☑ Ligações ☑ Apoios☑ Cargas ☑ Níveis ☑	Geometria Materiais Ligações Apoios Cargas Níveis	T T T T P T	Geometria Materiais Ligações Apoios Cargas Níveis	T P T* N N P
C	Robot	Revit Architecture	Geometria ☑ Materiais ☑ Ligações ☑ Apoios ☑ Cargas ☑ Níveis ☑	Geometria Materiais Ligações Apoios Cargas Níveis	T T N T N T	X	X
D	Robot	Revit Structures	Geometria ☑ Materiais ☑ Ligações ☑ Apoios ☑ Cargas ☑ Níveis ☑	Geometria Materiais Ligações Apoios Cargas Níveis	T N T T N T	X	X
E	Revit Structures	Revit Structures Archicad	Geometria ☑ Materiais ☑ Ligações ☑ Apoios ☑ Cargas ☑ Níveis ☑	/	-	Geometria Materiais Ligações Apoios Cargas Níveis	T P N N N P
F	Archicad	Tricalc	Geometria ☑ Materiais ☑ Ligações ☑ Apoios ☒ Cargas ☒ Níveis ☑	/	-	Geometria Materiais Ligações Apoios Cargas Níveis	T P T* S S P
G	Tricalc	Archicad Revit Robot	Geometria ☑ Materiais ☑ Ligações ☑ Apoios ☑ Cargas ☑ Níveis ☑	/	-	Geometria Materiais Ligações Apoios Cargas Níveis	T P N N N P
H	Tricalc	Tricalc	Geometria ☑ Materiais ☑ Ligações ☑ Apoios ☑ Cargas ☑ Níveis ☑	/	-	Geometria Materiais Ligações Apoios Cargas Níveis	T P P N N T
I	IFC4	Revit Robot Archicad Tricalc	Geometria ☑ Materiais ☑ Ligações ☑ Apoios ☑ Cargas ☑ Níveis ☑	X	X	/	-

Informação na Origem: ☑ - Introduzida; ☒ - Não Introduzida
Classificação dos níveis de interoperabilidade da informação: **T** – Transferida; **P** – Parcialmente transferida ou modificada; **N** – Não transferida; **S** – Sem aplicação prática; **X** – Incompatibilidade;
/ – Não aplicável; ***** – Informação adicionada pelo programa destino.

4

MODELAÇÃO EM IFC (CASO DE ESTUDO)

4.1. DESCRIÇÃO SEMÂNTICA

Uma linguagem é definida usando um conjunto de expressões formadas de acordo com um conjunto de regras, o que define a gramática. Conhecendo-se os requisitos gramaticais, o conjunto de regras é inesgotável, permitindo que a partir do conceito teórico uma linguagem tenha a sua própria composição e definição estrutural pelo arranjo dos seus componentes, formando uma sequência de informação inteligível. No domínio da linguagem de programação, a descrição de como a linguagem é composta e quais são os seus constituintes é definida pela sua sintaxe e semântica, isto é, pelas regras de construção de frases e seu significado, que devem ser especificadas com precisão. A sintaxe é vista como uma compilação livre de contexto, sendo a semântica incumbida de optimizar o código e fazer a verificação dos tipos. A semântica de uma linguagem de programação e o seu mapeamento é compilada pelo Homem, criando o que se designa muitas vezes por linguagem de código (ou apenas código), cuja finalidade é a execução computacional, com especial vocação para a repetição e sistematização de acções. (Gunter 1992).

No caso da linguagem utilizada pelo padrão IFC, o desenvolvimento e manutenção fica a cargo do "*Model Support Group*" (MSG), grupo implementado pela BuildingSMART. Para que uma linguagem permita a comunicação entre o emissor e o receptor, ambos devem comunicar da mesma forma, na medida em que o receptor seja capaz de reproduzir com exactidão as informações que lhe foram emitidas. Aplicando estes conceitos ao caso de estudo em causa, poder-se-á dizer que quando o emissor e o receptor não comunicam na mesma linguagem, o IFC funciona como tradutor, permitindo um entendimento entre a fonte e o destino. Esta é a finalidade que fomenta o IFC, a possibilidade de troca de informação por meio de uma linguagem comum (Pazlar and Turk 2008).

4.1.1. OBJECTIVOS

O objectivo principal deste capítulo é descrever a construção das estruturas que foram modeladas a partir de arquivos de textos simples para criação dos ficheiros IFC (que também foram usados nos testes de interoperabilidade do Capitulo 3), isto é, mostrar o encadeamento da linguagem IFC quando este é o próprio emissor. São dadas breves explicações sobre as definições e termos usados, para facilitar a compreensão e relações entre as classes IFC. Uma vez que o estudo do IFC é extremamente vasto, pretende-se com este capítulo demonstrar a elaboração de um arquivo IFC que poderá futuramente ser exportado e usado em softwares diversos. Este capítulo pretende então representar um

guia passo-a-passo que facilite e permita a continuação do estudo iniciado, por parte de investigações futuras.

4.1.2. Perspectivas Futuras

Nos dias que correm, as tecnologias de informação (TIC) evoluem com elevada celeridade e os caminhos que seguem estão constantemente associados à mudança e à incerteza, tornando difícil determinar com exactidão o que se tornará corrente e o que cairá no domínio do obsoleto pelo que, é necessário acompanhar o evoluir das ferramentas e dos processos tecnológicos o mais próximo possível. Seguindo esta filosofia, de entre os vários esquemas IFC escolheu-se o último esquema IFC4 como esquema de base para a modelação das estruturas elaboradas. Este esquema é ainda muito recente (lançamento oficial em Março de 2013) e por isso existem ainda muitos entraves à sua plena utilização, pois a maioria dos softwares, incluindo os mais usados no mercado, ainda não reconhecem este novo esquema, pelo que de momento ainda não é passível de utilização nessas ferramentas. Contudo e apesar das várias limitações e barreiras que existem actualmente na aplicação do esquema IFC4, entendeu-se que na conjuntura da presente dissertação o uso de um esquema IFC2x3 não permitiria um grande acréscimo de conhecimento para além do que já existe documentado em outras dissertações. A normalização deste novo esquema IFC4 segundo uma norma ISO (nomeadamente a ISO 16739:2013) e a sua especial vocação para o ramo estrutural, são pontos fortes seu a favor que justificam a sua utilização e que constituem também indicadores de que num futuro próximo este esquema possa ultrapassar a maioria das limitações dos esquemas anteriores. Estas limitações referidas estão na sua maioria relacionadas com o facto de que na situação actual a informação que é enviada para o STEP-file (vulgarmente chamado ficheiro IFC) gerado pelos programas sofre uma espécie de selecção e triagem da informação, perdendo-se informações durante o processo de troca, anulando grande parte do potencial associado ao IFC.

A confirmarem-se estas previsões, ter-se-á uma entidade externa a todos os programas que será verdadeiramente interoperável e de grande viabilidade, tornando o IFC num caminho quase incontornável. Entende-se por isso que o IFC4 será um estímulo para o uso desta linguagem e uma mais-valia para toda a indústria AEC (Arquitectura, Engenharia e Construção).

4.1.3. Implementação do IFC4

O surgimento do novo esquema IFC4, formalmente conhecido como IFC2x4, abrange ainda uma maior área de trabalho com uma série de novos recursos que contribuem activamente para uma melhoria da especificação IFC. O objectivo geral é aumentar a coerência em todo o esquema IFC, aplicando determinadas mudanças que resultaram da aprendizagem continua e do uso actual, tornando-o mais consistente e reduzindo o número de instâncias necessárias para a definição de determinado tipo de objecto. Como descrito pela BuildingSMART, o IFC4 “foi desenvolvido como sendo a próxima base para o IFC que permitirá a interoperabilidade dos *Building Information Models* e um *standard* para o openBIM” e foi reforçado através da norma ISO 16739:2013 que faz a definição do formato e estrutura de troca BIM que é necessária durante o ciclo de vida e fases de construção de um edifício, bem como os requisitos das várias disciplinas envolvidas. A norma inclui informações sobre a estrutura do projecto, componentes físicos, componentes espaciais, itens de análise, processos, recursos, controlos, intervenientes e definição do contexto.

Foi implementado um novo conceito de biblioteca onde são definidos todos os tipos de objectos, famílias ou estilos, e seus atributos que possibilitam a definição de modelos de propriedades a serem

incluídos num mesmo conjunto de dados do projecto. Os tipos de produtos podem agora ser decompostos em partes declaradas, que podem ser usadas para a comunicação entre as partes de uma ocorrência decomposta, sendo também suportado o conceito de cópia de um determinado tipo de produto para implementação numa estrutura de partes decompostas. Agora a árvore hierárquica das entidades, que faz a definição das ocorrências e dos tipos de objectos, para além de estar sincronizada com os seus conceitos e definições está também sincronizada com tipos de enumerações pré-definidos que fazem uma maior especificação das propriedades, nas entidades em que este conceito é aplicável. Esta árvore hierárquica das entidades pode classificar-se como um catálogo de objectos IFC, que faz o encadeamento de uma listagem de entidades e suas relações com sub listagens de definições, permitindo deste modo fazer um mapeamento externo completo. Na implementação do IFC4 as definições de componentes de serviços e sistemas (do qual o estrutural faz parte), sofreram uma melhoria significativa que consistiu na extensão do âmbito e adopção de uma metodologia de implementação de entidades de forma mais sistemática. As melhorias implementadas abrangeram os elementos de construção, elementos espaciais, construção de sistemas e serviços, cálculos de energia, elementos AVAC, sistemas eléctricos, modelação estrutural e detalhamento de elementos, análise estrutural de elementos, definição de novas propriedades, representação geométrica, sistemas de coordenadas, processos, custos, recursos de construção, restrições e aprovações, data e tempo, materiais, classificações e especificações de uso geral. Assim, este catálogo faz parte activa da indústria AEC (Figura 4.1), permitindo a definição de estruturas de troca que podem suportar e auxiliar as diferentes vocações.

Figura 4.1 – Sequência de palavras-chave relacionadas com o IFC

Dado o vasto domínio do IFC, a presente dissertação incidiu sobretudo sobre as áreas de modelação estrutural, análise estrutural e representação geométrica. A maioria dos subtipos dos elementos da construção (tais como sapatas, pilares, vigas, lajes, portas, janelas, etc.) foram separados numa definição geral e especialização específica para representação dos casos *standard*, possibilitando a troca de dados de forma, materiais e do elemento base que resulta na já mencionada metodologia de implementação sistematizada. As grandezas de base padronizadas estão agora adicionados à especificação IFC e válidas para toda a construção. Foram também implementados novos "*Property Sets*" para capturar o ambiente através de indicadores e foram adicionados valores para os elementos e tipos de elementos. Dadas as alterações nesses elementos de construção, a definição de lajes, vigas e pilares e elementos similares tem sido reforçada. A secção geométrica e informação dos materiais é agora associada aos elementos de construção. As definições da secção estão agora contempladas num nível mais alto que a representação geométrica, podendo também ser adicionadas referência a normas ou bibliotecas e também uma série de detalhes. No domínio do modelo de análise estrutural as

alterações e melhorias efectuadas foram no sentido de fazer a simplificação dos processos. O uso da representação topológica e sistema de coordenadas clarificado, bem como uma associação de materiais e secções atribuídas pelos elementos de construção, tornam o modelo mais flexível e versátil. Para além da aplicação de cargas é também agora possível a atribuição das forças resultantes e condições de fronteira (BuildingSMART 2013). Poder-se-á encontrar no site da BuildingSMART uma tabela com informação e lista detalhada das entidades que sofreram alterações na transição do esquema IFC2x3 para o esquema IFC4.

4.2. Caso de Aplicação

4.2.1. Proposta

O modelo proposto para modelação a partir da linguagem de código do IFC é um modelo que constitui na sua essência, informação de definição analítica de pilares, vigas e lajes estruturais e informação de definição geométrica de sapatas, pilares, vigas e lajes. Para além disso contempla também informação mecânica referente à definição de algumas propriedades físicas dos materiais e casos de aplicação de diversos tipos de cargas à estrutura modelada.

Procurou-se então elaborar um modelo de uma estrutura (Figura 4.2) que de forma simplista fosse capaz de captar todas estas características, para dessa forma ser facilmente interpretável e de fácil exposição. A estrutura porticada elaborada assemelha-se a um anexo ou uma garagem simples, com uma definição geométrica puramente académica não tendo sido realizados nenhum tipo de pré-dimensionamento. A estrutura é composta por quatro pilares de secção 0,20x0,30m e 3m de altura implantados numa malha quadrada cujos vértices distam entre si 5,50 metros e apoiados sobre sapatas 1,50x2,00x0,75m. As vigas em ambas as direcções têm uma secção de 0,20x0,40m, tendo o par de vigas na direcção X uma extensão face a face de 5,20 metros e na direcção Y uma extensão de 5,30 metros. Esta diferença de extensões deve-se à diferença entre as dimensões das faces dos pilares (já que estes são rectangulares). A laje tem uma espessura de 0,22m e as suas dimensões correspondem ao perímetro interior formado pelos pilares e pelas vigas, logo 5,30x5,40m.

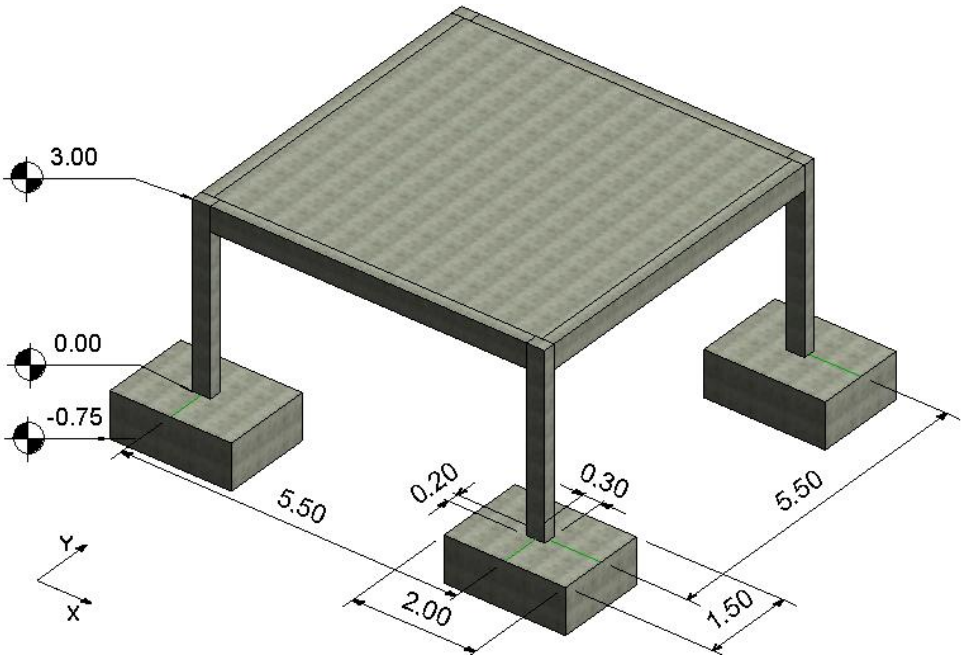


Figura 4.2 – Dimensões de implantação da estrutura porticada.

A definição analítica é representada por nós e barras e por isso é feita basicamente pela colocação de vértices na mesma malha quadrada (5,50x5,50x3,00m) e pela união desses vértices por barras que representam os pilares e vigas da estrutura. É possível associar secções a esses modelos analíticos ou até mesmo elementos geométricos.

Quanto ao material atribuído aos elementos foi o betão C20/25, do qual se definiram algumas características como classe estrutural, classe de exposição, classe de resistência, recobrimento, etc. e propriedades físicas como a densidade, módulo de cisalhamento e módulo de elasticidade. Outras particularidades poderiam ter sido atribuídas, mas não era o objectivo principal uma definição exaustiva dos materiais.

Relativamente às cargas aplicadas à estrutura foram definidos diferentes tipos de forças para demonstração de alguns casos de carga. Foi então aplicada uma carga pontual, uma carga linear uniformemente distribuída, uma carga linear com distribuição trapezoidal e uma carga superficial uniformemente distribuída.

4.2.2. Elaboração do Modelo IFC

A elaboração do modelo criado foi dividido essencialmente em duas partes, uma que define a estrutura enquanto modelo analítico e outra que define a estrutura do modelo geométrico. Infelizmente com a limitação de acesso ao *software* "Constructivity" não permite a visualização do modelo geométrico e sendo que este foi o único programa capaz de abrir o esquema IFC4 a construção do modelo teve também de ser dividida nestas duas partes referidas. A parte que dizia respeito ao modelo analítico foi elaborada no esquema IFC4 e a verificação visual feita no programa "Constructivity Model Viewer" (Figura 4.3), a parte que dizia respeito ao modelo geométrico foi feita no esquema IFC2x3 e a verificação visual feita no programa "Solibri Model Viewer" (Figura 4.4) e só depois então é que se fez a transposição desse modelo geométrico criado para o mesmo ficheiro IFC4 onde estava o modelo analítico (com as devidas ressalvas devido às alterações que foram feitas em algumas entidades na transição do esquema IFC2x3 para o IFC4, também já mencionadas anteriormente). O código completo da linguagem usada no modelo IFC poderá ser encontrado em anexo.

Faz-se aqui uma chamada de atenção relativamente à definição analítica da laje estrutural pois não está correctamente definida, como se pode ver pela Figura 4.3 a laje aparece representada à cota zero ao invés de estar representada à cota três metros como pretendido. A representação da carga estrutural da laje também é definida à cota zero pois está associada ao elemento estrutural da laje, apenas não se tornou essa visualização activa para que se pudesse identificar a laje no terreno e para não complicar a visualização das restantes cargas. A definição analítica revelou-se bastante complexa e o motivo desta definição incorrecta resulta da limitação do tempo que houve para o estudo destas matérias, não permitindo aprofundar os conhecimentos em todos os elementos.

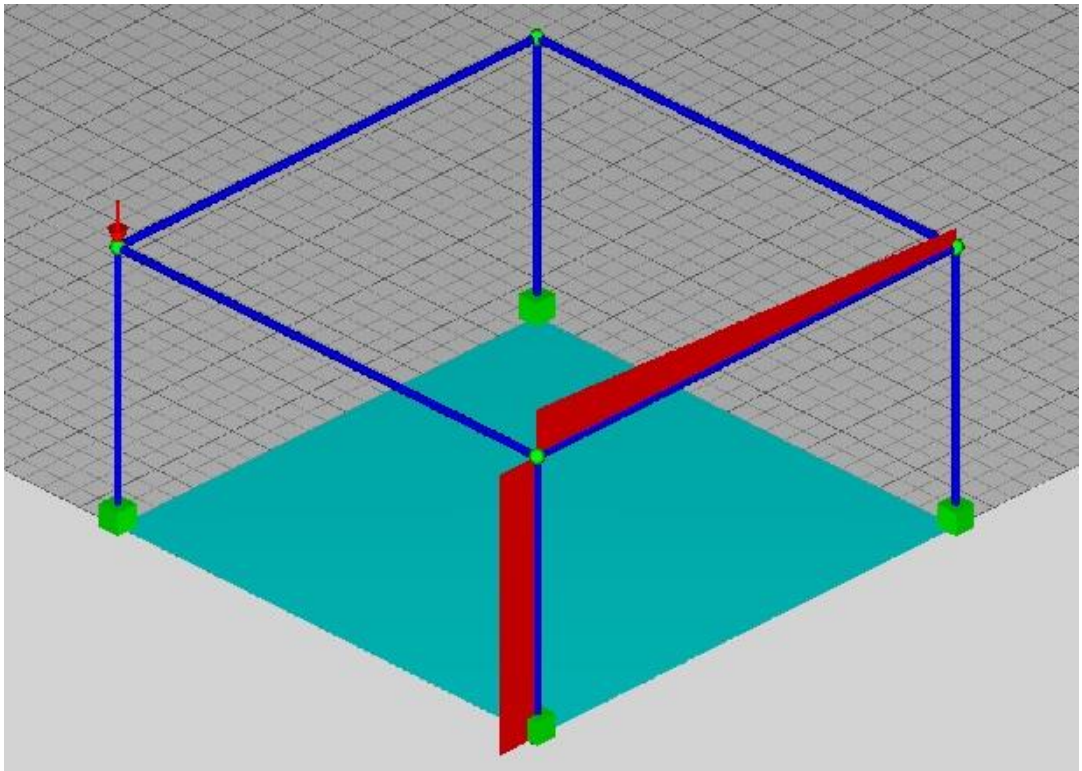


Figura 4.3 – Visualização do modelo IFC analítico através do Constructivity Model Viewer

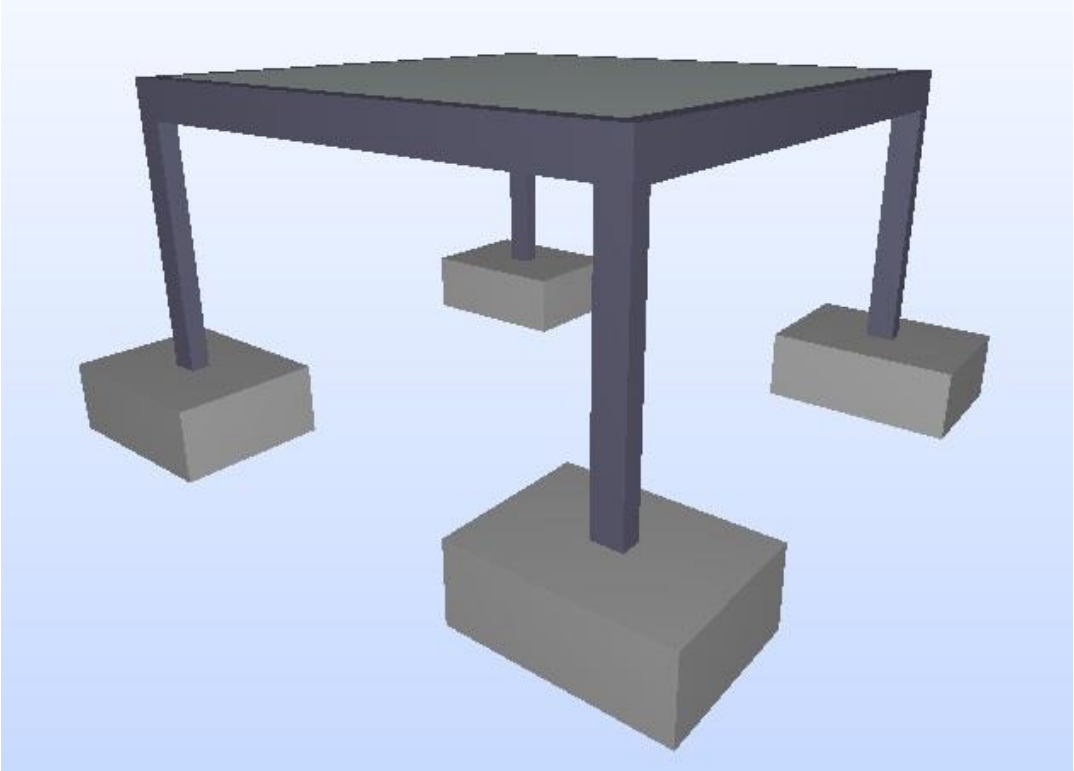


Figura 4.4 – Visualização do modelo IFC analítico através do Solibri Model Viewer

4.3. ESTRUTURA INTERNA DE FICHEIROS IFC

Para além de validada pela experimentação prática, a exposição e demonstração de aplicação que será feita, teve como suporte base a ISO 10303-21:2002 e o site da BuildingSMART.

4.3.1. CONCEITOS BASE

Antes de se abordar a construção dos modelos no formato IFC, deve haver um conhecimento mínimo sobre os termos e definições que lhe estão associados. Uma vez que já foram oportunamente expostos no Capitulo 1 alguns conceitos associados ao IFC, importa agora introduzir o modo de funcionamento de um ficheiro IFC.

A primeira linha e última linha de todos os arquivos IFC devem identificar a norma ISO 10303 cujo objectivo é estabelecer procedimentos e linguagem padrão para representação, interpretação e troca de informação, sendo que a primeira linha deve conter a referência "`ISO-10303-21`" e a última a referência "`END-ISO-10303-21`". Esta referência evidência o formato de um arquivo STEP definido segundo a ISO 10303-21:2002, onde o formato de troca de dados STEP mais utilizado é o STEP-File, que devido à sua estrutura ASCII é fácil de ler com tipicamente uma instância por linha. Denote-se ainda que a ISO 10303-21 define o mecanismo de codificação na forma de representar os dados de acordo com um determinado esquema EXPRESS, padrão de linguagem para modelação de dados, baseando-se no EXPRESS mas constituindo uma variante. (ISO 2002).

O corpo de um arquivo que segue esta norma é dividido em duas secções, a secção "*HEADER*" e a secção "*DATA*". Para se dar como concluída cada uma destas secções é usada uma *string* com o comando "*ENDSEC;*". A Figura 4.5 demonstra a forma de elaboração genérica de uma estrutura de troca de informação de um ficheiro STEP, como foi supracitado.

```
ISO-10303-21;

HEADER;
…
ENDSEC;

DATA;
…
ENDSEC;

END-ISO-10303-21;
```

Figura 4.5 – Estrutura tipo de um ficheiro STEP (Nour 2008)

A secção HEADER tem uma estrutura fixa, podendo ser constituída por 6 grupos, sendo que todas as estruturas de troca devem conter os grupos "*FILE_DESCRIPTION*", "*FILE_NAME*", "*FILE_SCHEMA*" e devem aparecer na ordem pela qual serão abaixo descritos. Instâncias dos grupos "*FILE_POPULATION*", "*SECTION_LANGUAGE*" e "*SECTION_CONTEXT*" podem aparecer depois do "*FILE_SCHEMA*" apenas em arquivos de segunda edição. Nesta secção todos os campos podem incluir *strings* vazias, excepto o grupo "*FILE_SCHEMA*" e os dados no campo "*time_stamp*" que está incluído no grupo "*FILE_NAME*" (ISO 2002).

- `FILE_DESCRIPTION` - atribui informação acerca da definição de visualização base e também o nível de implementação, que identifica a especificação com a qual a codificação na estrutura

de troca conforma (correntemente "2,1" para todos os ficheiros IFC, para mais informações recomenda-se a consulta da ISO 10303-21).

- `FILE_NAME` - regista o nome do ficheiro IFC (podendo corresponder ao nome do arquivo num sistema de arquivos ou retractar os dados nesse arquivo), a data de criação do ficheiro (o tempo é definido no formato internacional segundo a ISO 8601), o nome do autor e organização a que pertence, a aplicação na qual o ficheiro foi criado e autorização (nome e endereço da pessoa que autorizou o arquivo).

- `FILE_SCHEMA` – Especifica um esquema EXPRESS ou vários no caso de arquivos de segunda edição. Um esquema é um modelo de dados legível por um sistema informático, que descreve conjuntos de tipos de dados e suas relações.

- `FILE_POPULATION` – Indica uma população válida (conjunto de instâncias de entidade) em conformidade com o esquema EXPRESS. Isto é feito através de recolha de dados a partir de várias secções de dados.

- `SECTION_LANGUAGE` – Identifica a linguagem predefinida para os valores das *strings* na secção dos dados. O nome da linguagem deve aparecer codificado segundo a norma ISO 639, no campo de atributo *default_language*. Isto é necessário para os esquemas EXPRESS que não possuem a capacidade de especificar em que língua estão definidas as entidades e os seus atributos.

- `SECTION_CONTEXT` – Identifica informações que descrevem os contextos nos quais são aplicáveis as instâncias codificadas na estrutura de troca.

A secção DATA contém dados de aplicativos de acordo com um esquema EXPRESS específico, é nesta secção que as entidades são atribuídas ao ficheiro IFC. Quando se trabalha directamente com o esquema EXPRESS numa ferramenta de edição de texto, é inevitável falar-se em entidades (ou classes), pois no domínio IFC, são as entidades que definem os comandos e acções capazes de criar informação.

Uma entidade tem uma estrutura bastante própria, mostrada na Figura 4.6 e exemplificada na Figura 4.7. É reconhecida no ficheiro IFC pelo seu identificador único que deve ser um número inteiro positivo (> 0), na forma "#número". A finalidade e função dessa entidade é identificada pelo seu nome, que é representado por letras maiúsculas no código IFC. Repare-se que o nome dessa entidade é apenas válido dentro desse ficheiro IFC, pois se o conteúdo deste for exportado para um novo sistema, mesmo que não se efectuem quaisquer alterações, ao voltar-se a exportar o ficheiro para o IFC a partir desse sistema, verifica-se que o código do ficheiro é alterado, ou seja, há uma alteração e modificação das entidades para as mesmas instâncias. O nome da instância é usado também para referenciar outras entidades através de valores atribuídos ou membros agregados (atributo inverso), sendo que a instância referenciada poderá ser definida antes ou depois da instância onde é referida. Por fim as definições de atributos atribuem características à entidade IFC em causa. Os valores e definições desses atributos variam de entidade para entidade, pois definem parâmetros próprios para cada uma das diferentes entidades. Como já foi referido, esses parâmetros são definidos a priori pelo MSG (*Model Support Group)*. Uma lista completa das entidades existentes, com *links* individuais que fazem a respectiva ligação à página que especifica a função, construção e relações de cada entidade pode ser encontrada em:

http://www.buildingsmart-tech.org/ifc/IFC4/final/html/annex/annex-b/alphabeticalorder_entities.htm

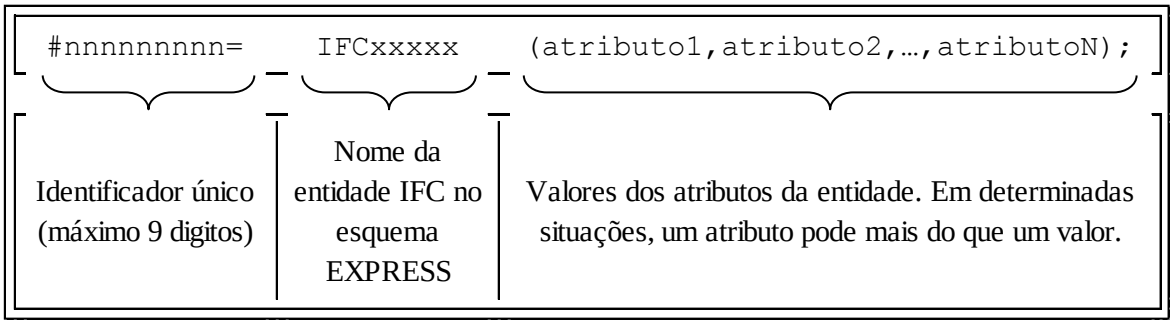


Figura 4.6 – Organização de uma entidade. Adaptado de (Nour 2008)

```
#224= IFCSTRUCTURALPOINTCONNECTION ('2GClK7cwT80xzpZuaAlGXp',#5,'Point
                                     Connection 1',$,$,$,#223,#225,$)
```

Figura 4.7 – Exemplo de uma entidade.

A sequência pela qual as entidades aparecem descriminadas na construção de um dado texto em código não corresponde obrigatoriamente à ordem pela qual serão identificadas pelo programa destino, isto significa que, em duas linhas de código consecutivas, a segunda *string* não tem necessariamente que depender ou ter alguma relação com a primeira. O IFC é um sistema de digitação solto que fornece múltiplos caminhos para introdução de objectos e não segue uma orientação específica, permitindo a criação de vários encadeamentos lógicos, dando assim grande flexibilidade e suporte a múltiplas representações (ISO 2002).

A título de exemplo, um conjunto de pilares não têm necessariamente que estar definidos por alguma ordem em concreto, isto permite grande versatilidade na introdução ou remoção de entidades consoante as necessidades do utilizador, para além disso, os pilares podem partilhar a mesma secção, material ou outras características que possam ter em comum, possibilitando assim uma eficiência na utilização e organização de entidades que resulta numa diminuição do tamanho do ficheiro IFC.

Então, na construção de um código IFC não existe um único caminho ou uma sequência obrigatória, os caminhos de leitura das entidades são definidos pela sua indexação e hierarquização dentro de uma camada e pelas suas relações de atributos. Portanto o IFC é dividido em camadas, aliás como já foi apresentado no Capitulo 2 (Domínios → Elementos Partilhados → Nuclear → Recursos) e é dentro da camada de "recursos" que se encontram os esquemas de dados que suportam as estruturas de dados do modelo. As entidades mais genéricas encontram-se dentro dessa camada, mas podem ser referenciadas por todas as entidades do modelo. Como o IFC define relações entre as entidades baseado no formato EXPRESS, as centenas de entidades são organizadas numa cadeia hierárquica baseada nos objectos. As entidades são separadas genericamente em dois tipos, as que derivam directamente do *IfcRoot* (identificadas pelo GUID e com possibilidade de atribuição de *IfcOwnerHistory*, nome e descrição) e as que existem se forem associadas directa ou indirectamente às primeiras (não são identificadas através do GUID).

Para que um objecto possa ser identificado quer pelo operador quer por um sistema de processamento automático, necessita de um identificador. A especificação IFC usa um identificador único para instâncias de objectos denominado *Globally Unique Identifier* (GUID) que segue o Universal Unique Identifier Standard (UUID) como definido na ISO/IEC 11578:1996. O GUID gerado para uma instância do ficheiro de troca é comprimido segundo uma fórmula de compressão que tem como base

uma codificação de 64 caracteres (Figura 4.8) e que resulta num identificador de 22 caracteres, podendo a sua introdução ser automática ou manual. Permite fazer a identificação de um objecto independentemente da aplicação que está a ser usada, sendo desse modo possível a sincronização entre programas que potencialmente oferece a possibilidade de alteração dos aspectos dos objectos por parte de outros utilizadores, ou seja, desse modo é possível coordenar os diferentes níveis de objectos em diferentes sistemas. (Eastman et al. 2011).

```
          1         2         3         4         5         6
0123456789012345678901234567890123456789012345678901234567890123
"0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz_$";
```

Figura 4.8 – Codificação base de 64 caracteres para a formulação de um IFC-GUID. (BuildingSMART 2013)

Resumidamente, para evitar possíveis confusões entre as funções de um GUID e de um Identificador Único, o GUID permite a identificação de objectos nas relações de interoperabilidade entre sistemas, mantendo-se inalterado na realização dessas trocas e permitindo dessa forma a identificação de elementos na estrutura da linguagem IFC, como demonstrado na Figura 4.9. O identificador único de uma entidade apenas é reconhecido dentro dessa estrutura de código de um dado ficheiro IFC, sendo usado para fazer nomear entidades e fazer relações por atributos (Firmenich et al. 2006).

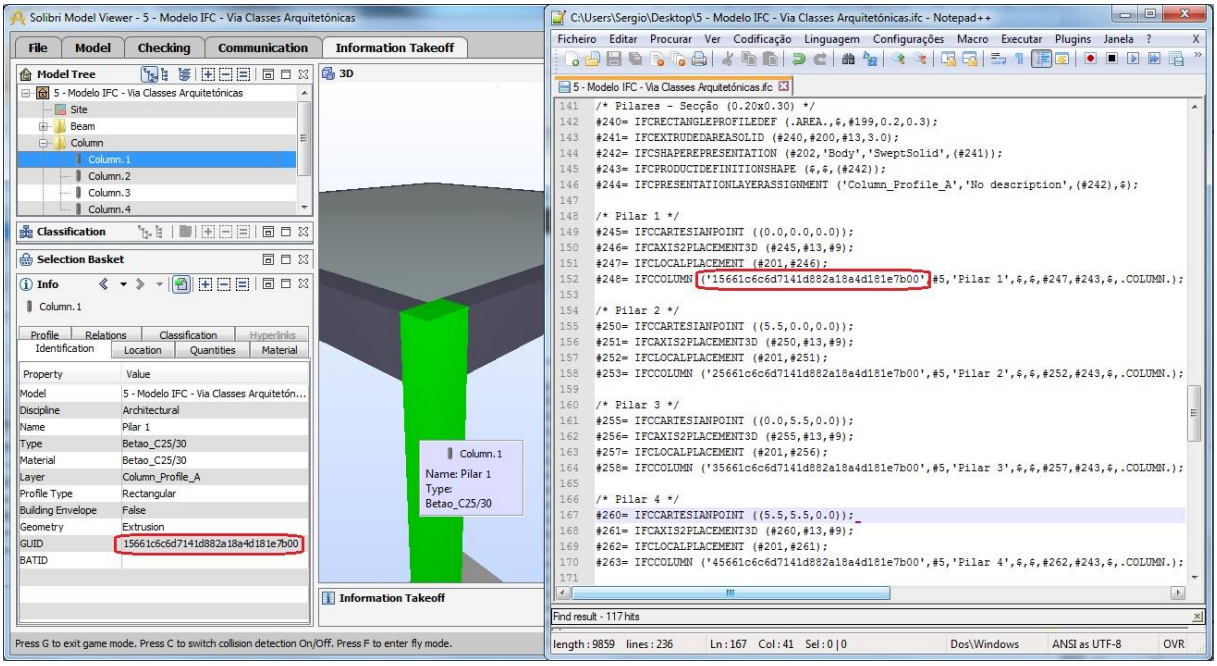


Figura 4.9 – Correspondência entre o GUID no ficheiro IFC e no visualizador Solibri

O *IfcRoot* é a classe mais abstracta e a raiz de todas as definições das entidades IFC, sendo o supertipo de forma directa do *IfcObjectDefinition*, *IfcPropertyDefinition* e *IfcRelationship* (são denominadas classes subtipo do *IfcRoot*), entidades que por sua vez também constituem o supertipo de outras entidades e assim sucessivamente. Resumidamente é através desta identificação e relação entre as entidades supertipo e subtipo e pela classificação dos atributos (directo, inverso e derivado) que se

definem as cadeias de classes e o seu seguimento lógico, possibilitando a leitura do código pelo programa destino (BuildingSMART 2013).

Tem-se então um modelo IFC que segue uma filosofia baseada nos *Semantic Exchange Modules* (SEM), que se apresenta como um modelo de desenvolvimento por módulos destinado a facilitar a visualização do modelo e também a reutilização das entidades. Um SEM é um subconjunto modular estruturado de objectos e relações necessárias para definir cada um dos múltiplos modelos de troca BIM. O seu objectivo é permitir que as empresas de software na área dos BIM consigam importar e exportar informação de código através de módulos, para que uma dada informação ao ser exportada ou importada possa ser testada e certificada uma vez e depois possa ser reutilizada em múltiplos ficheiros de troca sem que haja necessidade de modificação. Um SEM pode ser visto como uma ligação a um conjunto de entidades IFC, seus atributos, relações e funções. O âmbito do SEM deve ser determinado conjuntamente com a equipa que desenvolve o software, visto que devem fazer atribuições não só ao modelo de troca, mas também aos objectos internos do esquema da ferramenta. A semântica nas áreas de engenharia e design é particular no sentido que define uma mistura entre especificações parciais da realidade, comportamentos e funcionamento expectáveis dessa realidade e os seus próprios sistemas físicos. A semântica é também necessária para fazer a distinção entre definições e objectos dentro de um domínio. O IFC oferece um esquema que faculta instâncias de especificações quer dos projectos de edifícios, quer dos edifícios reais. Os edifícios são constituídos por distintos sistemas de diferentes vocações cuja definição obriga à existência das suas próprias entidades, bem como entidades partilhadas. Isto implica e significa que há mais do que uma forma de representar determinada informação num ficheiro de troca. Várias relações têm como domínio o *IfcRoot*, que é bastante amplo e não restringe o uso das relações. Já é vasto o conjunto de entidades que fazem parte do IFC, mas existem áreas de trabalho que ainda não foram identificadas no IFC em forma de classes, tendo esse trabalho sido deixado para esquemas IFC futuros. Para se ultrapassar esta situação é necessário amplificar o uso do IFC e é nesse sentido que surge a normalização do IFC4, com o intuito de tornar o IFC ainda mais aplicável e num âmbito mais alargado. (Venugopal 2011).

4.3.2. Simbologia Usada

Existem alguns caracteres especiais usados na construção de ficheiros IFC, aos quais são atribuídas funções específicas (Lehrenfeld, Mueller, and Wiechers 1993), os principais usados são:

- # – Usado antes de um número inteiro para que esse número seja definido como o identificador único de uma entidade.
- $ – Valor indefinido.
- * – Atributo omisso.
- , – A vírgula faz a separação entre os diferentes atributos de uma dada entidade. Para representação de números decimais não se usa a vírgula “,” mas sim o ponto “.” .
- `''` – O conteúdo definido entre dois apóstrofos é geralmente texto alfanumérico, normalmente elaborado para ser facilmente interpretado.

  Exemplo: `#202= IFCGEOMETRICREPRESENTATIONCONTEXT ('3D','Model',3, 0.00001,#200,$);`

- . . – Um atributo definido entre dois pontos representa um atributo do tipo enumeração e deve ser escrito em letras maiúsculas. Se atribuído, deve corresponder obrigatoriamente a uma

enumeração pré-definida. As enumerações respectivas a uma entidade (no caso desta necessitar de alguma) podem ser encontradas em http://www.buildingsmart-tech.org.

Exemplo: `#5= IFCOWNERHISTORY (#4,#2,.READWRITE.,.NOCHANGE.,$,$,$, 1320677205);`
http://www.buildingsmart-tech.org/ifc/IFC4/final/html/schema/ifcutilityresource/lexical/ifcstateenum.htm

- ( ) – Os parêntesis delimitam os atributos de uma dada entidade. Se reutilizados dentro da própria secção de atributos, possibilitam a colocação de um ou mais valores num único atributo de uma entidade.

Exemplo: `#190= IFCPROJECT ('0QjBRF7yLCTh4HRF3GOF1t',#6,'Project',$,$, $,$, (#202,#203),#160);`

- `/* */` - Permite a adição de comentários, observações ou anotações ao texto em linguagem IFC. As informações contidas entre `/* exemplo */`, não têm significado para o código e por isso não são lidas pelos programas de destino.

Exemplo: `/* Dados do usuário, organização e da aplicação */`

4.3.3. Análise e Exposição da Criação de um Ficheiro IFC

Depois de introduzidas as considerações iniciais que devem estar presentes quando se trabalha com a linguagem IFC, é feita agora a análise da criação de um ficheiro IFC, cuja exposição será feita modularmente. Note-se que a organização que foi atribuída às entidades que constituem o ficheiro IFC em nada influência ou altera a sua leitura por parte dos softwares, como já foi mencionado a organização do ficheiro IFC é feita pela hierarquização das entidades, não pela sua ordem no texto de código. Este arranjo na organização do código do ficheiro IFC apenas serve para tornar a linguagem humanamente mais legível, facilitando a leitura ao utilizador. Mais uma vez se refere que a modelação em linguagem IFC teve como base principal o grupo de suporte e implementação do modelo IFC da BuildingSMART que inclui uma biblioteca de entidades e suas relações, tornando assim possível encontrar todas as informações necessárias para modelar os elementos estruturais que serão apresentados. Essa biblioteca de entidades está publicada no formato HTML, com listas de todos os tipos definidos, enumerações e tipos de entidades, listadas por ordem alfabética, compiladas num documento de navegação, com uma organização em árvore segundo que constitui uma lista hierárquica entre supertipos e subtipos.

Refere-se novamente que a investigação e pesquisa das entidades para a definição da estrutura porticada que será exposta, foi usado o programa "Constructivity" com o qual se foi abrindo periodicamente o ficheiro IFC do modelo que se estava a criar, para se ter a certificação de que o código estava a ser correctamente implementado. A versão do programa que foi utilizada foi uma versão de licença temporária com inúmeras funcionalidades indisponíveis, sendo que foi apenas utilizado como visualizador do modelo analítico da estrutura. Para fazer a visualização do modelo arquitectónico criou-se um modelo IFC2x3 em paralelo ao original, contendo apenas as classes arquitectónicas que deste modo poderia ser visualizado com o programa "Solibri Model Viewer" e fazer-se assim a verificação de todo o modelo.

O exemplo que se segue diz respeito a uma estrutura porticada. A preocupação central foi a definição estrutural dos elementos, mas também foi definido o modelo arquitectónico da estrutura, sendo depois feito o relacionamento entre classes estruturais e classes arquitectónicas.

4.3.3.1. HEADER

Como em todos os ficheiros IFC, o primeiro passo a tomar é construir ou preencher a secção HEADER, com informação geral acerca do ficheiro e do criador, ou seja, informação relevante para todo o ficheiro global, que deve ser atribuída segundo as indicações que foram já supracitadas. Serve de exemplo a Figura 4.10 e a Figura 4.11 que mostra a secção HEADER que faz parte do modelo IFC criado.

```
HEADER;
   FILE_DESCRIPTION (('ViewDefinition [CoordinationView]'),'2;1');
   FILE_NAME ('ModeloIFC.ifc','2013-04-03T10:54:01','Sérgio Pinho','FEUP',
              'BIM','Notepad++',$);
   FILE_SCHEMA (('IFC4'));
ENDSEC;
```

Figura 4.10 – Modelo IFC - Secção HEADER do modelo IFC criado.

```
HEADER;
   FILE_DESCRIPTION (("Descrição"),"Nível de implementação");
   FILE_NAME ("Nome","Data","Autor","Organização","Sistema","Aplicação",
              "Autorização");
   FILE_SCHEMA (("Esquema EXPRESS"));
ENDSEC;
```

Figura 4.11 – Aplicação - Secção HEADER do modelo IFC criado.

4.3.3.2. DATA – Dados do utilizador e da aplicação

Concluída a secção HEADER segue-se a secção DATA, secção onde se dispõe a linguagem de código propriamente dita. Por uma questão de metodologia, facilidade de trabalho e encadeamento lógico, as entidades discriminadas na secção DATA estão organizadas por módulos (grupos), pelo agrupamento de informação de natureza semelhante ou que contribui para uma mesma finalidade. O primeiro grupo introduzido no ficheiro IFC em questão contem as instâncias que introduzem as informações relacionadas com o usuário, dados da aplicação e também da organização (Figura 4.12 e Figura 4.13). Novamente refere-se que a secção DATA tem obrigatoriamente que iniciar-se com a *string* “DATA;”.

```
DATA;

/* Dados do usuário, da organização e da aplicação */
   #1= IFCORGANIZATION ($,'Sergio-PC','FEUP',$,$);
      #2= IFCAPPLICATION (#1,'1.0','IFC4','Made via Notepad++');
   #3= IFCPERSON ($,$,'Sergio',$,$,$,$,$);
      #4= IFCPERSONANDORGANIZATION (#3,#1,$);
         #5= IFCOWNERHISTORY (#4,#2,.READWRITE.,.NOCHANGE.,$,$,$,1320677205);
```

Figura 4.12 – Modelo IFC - Dados do usuário, da organização e da aplicação

```
#n= IFCORGANIZATION ("Identificação","Nome","Descrição","Cargo","Morada");
#n= IFCAPPLICATION ("Distribuidor da Aplicação","Versão","Nome da
                     Aplicação","Idenficador da Aplicação");
```

```
#n= IFCPERSON ("Identificação","Nome de familia","Nome próprio","Nome
                central","Título prefixo","Título sufixo","Cargo","Morada");
#n= IFCPERSONANDORGANIZATION ("Pessoa","Organização","Cargo");
#n= IFCOWNERHISTORY ("Utilizador proprietário","Proprietário da aplicação",
                     "Estado","Tipo de Modificações","Data da última
                      modificação","Utilizador na última modificação",
                    "Aplicação usada na última modificação","Data de criação");
```

Legenda: *IfcLabel*; *IfcIdentifier*; *IfcText*; *IfcTimeStamp*; *Enumeration*

Figura 4.13 – Aplicação - Dados do usuário, da organização e da aplicação

A Figura 4.13 fornece os dados necessários para uma definição completa das entidades usadas na Figura 4.12. Para facilitar a compreensão da exposição das entidades foram adoptadas algumas regras que serão usadas em todo o capitulo. As entidades com um maior avanço na linha são entidades que têm como atributo inverso uma outra entidade que está mais recuada, este princípio serve apenas para melhor demonstrar o encadeamento entre as classes num grupo. Os atributos marcados a sublinhado são atributos cuja atribuição é opcional, poderão ou não ser definidos (um atributo opcional é representado por um "$", valor indefinido). Também foi adoptado um esquema de cores que permitisse uma fácil e rápida identificação dos tipos de atributos mais usados. Os atributos do tipo *IfcText* contêm informação alfanumérica legível para o ser Humano com a função meramente informativa. Já os atributos do tipo *IfcLabel* são usados para definir o termo de algo a que se referem e representam também linguagem humanamente interpretável. O *IfcIdentifier* tem a função de identificar elementos ou objectos através de informação alfanumérica normalmente gerada pelo sistema informático e legível apenas para este, em contraste com o *IfcLabel* e o *IfcText*.

Tomando a classe "#2 = *IfcApplication*" como exemplo, nela podem ser encontrados atributos directos e um atributo inverso. Como atributo directo tem-se por exemplo a informação "*Made via Notepad++*" e como atributo indirecto tem-se "#1". Como se pode ver pela Figura 4.12 quando o número identificador de uma entidade é colocado dentro da secção de atributos de uma outra, é feita uma ligação entre entidades que funciona como uma espécie de *link*. Uma das entidades mais importantes deste grupo é a *IfcOwnerHistory*, que especifica informação relacionada com a identificação e historial da aplicação. Para tornar o seu acesso rápido, esta entidade poderá estar directamente indexada em todos os objectos independentes, relações e propriedades.

4.3.3.3. DATA – Pontos cartesianos e direcções

Recomenda-se que entre os vários grupos da secção DATA haja uma quebra numérica na numeração das entidades, desse modo no caso de surgir a necessidade de fazer a implementação manual de novas classes num determinado grupo é possível manter o ficheiro IFC arrumado.

O próximo passo foi colocar os pontos cartesianos que correspondem ao ponto de origem, em referenciais bidimensionais e tridimensionais, para futura colocação de eixos cartesianos na definição de objectos (Figura 4.14 e Figura 4.15). Também foram colocadas direcções que permitiram atribuir orientação a objectos segundo um determinado eixo, quer bidimensionalmente para orientação de planos, ou tradicionalmente para orientação de elementos.

```
/* Pontos cartesianos e direcções */
   #7= IFCCARTESIANPOINT ((0.,0.,0.));
```

```
#8= IFCCARTESIANPOINT ((0.,0.));
#9= IFCDIRECTION ((1.,0.,0.));
#10= IFCDIRECTION ((-1.,0.,0.));
#11= IFCDIRECTION ((0.,1.,0.));
#12= IFCDIRECTION ((0.,-1.,0.));
#13= IFCDIRECTION ((0.,0.,1.));
#14= IFCDIRECTION ((0.,0.,-1.));
#15= IFCDIRECTION ((1.,0.));
#16= IFCDIRECTION ((-1.,0.));
#17= IFCDIRECTION ((0.,1.));
#18= IFCDIRECTION ((0.,-1.));
```

Figura 4.14 – Modelo IFC – Pontos cartesianos e direcções.

```
#n= IFCCARTESIANPOINT (("Coordenada X"," Coordenada Y"," Coordenada Z"));
#n= IFCCARTESIANPOINT (("Coordenada X "," Coordenada Y"));
#n= IFCDIRECTION (("Direção X","Direção Y","Direção Z"));
#n= IFCDIRECTION (("Direção X","Direção Y"));
```

Figura 4.15 – Aplicação - Pontos cartesianos e direcções.

4.3.3.4. DATA – Unidades

Identificados os dados do usuário e da aplicação e os pontos na origem e direcções principais segundo os eixos coordenados, procede-se à definição das unidades que serão ou poderão ser usadas no ficheiro IFC (Figura 4.16 e Figura 4.17). A lista de unidades aplica-se a todos os valores de medida dentro do arquivo (é definida globalmente), por exemplo, a unidade de comprimento usada na geometria será o metro (#28), os ângulos serão medidos em graus (#43), o peso em quilogramas (#36), etc. As unidades podem ser classificadas como grandezas fundamentais (primitivas) ou derivadas. As grandezas fundamentais que não dependem de outras para serem definidas são sete e são elas o comprimento, massa, tempo, intensidade de corrente eléctrica, intensidade luminosa, temperatura termodinâmica e quantidade de matéria. As grandezas derivadas são definidas por relação entre grandezas fundamentais.

```
/* Unidades */
   #20= IFCSIUNIT (*,.AREAUNIT.,$,.SQUARE_METRE.);
   #21= IFCSIUNIT (*,.FORCEUNIT.,$,.NEWTON.);
   #28= IFCSIUNIT (*,.LENGTHUNIT.,$,.METRE.);
   #36= IFCSIUNIT (*,.MASSUNIT.,.KILO.,.GRAM.);
   #40= IFCSIUNIT (*,.PLANEANGLEUNIT.,$,.RADIAN.);
      #41= IFCMEASUREWITHUNIT (IFCPLANEANGLEMEASURE(0.0174532925199433),#40);
   #42= IFCDIMENSIONALEXPONENTS (0,0,0,0,0,0,0);
         #43= IFCCONVERSIONBASEDUNIT (#42,.PLANEANGLEUNIT.,'DEGREE',#41);
   #45= IFCSIUNIT (*,.PRESSUREUNIT.,$,.PASCAL.);
   #56= IFCSIUNIT (*,.VOLUMEUNIT.,$,.CUBIC_METRE.);
   #57= IFCSIUNIT (*,.TIMEUNIT.,$,.SECOND.);
   #58= IFCSIUNIT (*,.THERMODYNAMICTEMPERATUREUNIT.,$,.DEGREE_CELSIUS.);
   #59= IFCSIUNIT (*,.LUMINOUSINTENSITYUNIT.,$,.LUMEN.);
      #96= IFCDERIVEDUNITELEMENT (#21,1);
      #97= IFCDERIVEDUNITELEMENT (#28,-1);
         #98= IFCDERIVEDUNIT ((#96,#97),.LINEARFORCEUNIT.,$);
      #99= IFCDERIVEDUNITELEMENT (#21,1);
      #100= IFCDERIVEDUNITELEMENT (#28,1);
      #101= IFCDERIVEDUNITELEMENT (#28,-1);
         #102= IFCDERIVEDUNIT ((#99,#100,#101),.LINEARMOMENTUNIT.,$);
      #103= IFCDERIVEDUNITELEMENT (#21,1);
      #104= IFCDERIVEDUNITELEMENT (#28,-1);
         #105= IFCDERIVEDUNIT ((#103,#104),.LINEARSTIFFNESSUNIT.,$);
      #112= IFCDERIVEDUNITELEMENT (#36,1);
```

```
#113= IFCDERIVEDUNITELEMENT (#56,-1);
   #114= IFCDERIVEDUNIT ((#112,#113),.MASSDENSITYUNIT.,$);
 #118= IFCDERIVEDUNITELEMENT (#36,1);
#119= IFCDERIVEDUNITELEMENT (#28,-1);
   #120= IFCDERIVEDUNIT ((#118,#119),.MASSPERLENGTHUNIT.,$);
#121= IFCDERIVEDUNITELEMENT (#45,1);
   #122= IFCDERIVEDUNIT ((#121),.MODULUSOFELASTICITYUNIT.,$);
#140= IFCDERIVEDUNITELEMENT (#28,4);
   #141= IFCDERIVEDUNIT ((#140),.MOMENTOFINERTIAUNIT.,$);
#142= IFCDERIVEDUNITELEMENT (#21,1);
#143= IFCDERIVEDUNITELEMENT (#20,-1);
   #144= IFCDERIVEDUNIT ((#142,#143),.PLANARFORCEUNIT.,$);
#147= IFCDERIVEDUNITELEMENT (#36,1);
#148= IFCDERIVEDUNITELEMENT (#20,-1);
   #149= IFCDERIVEDUNIT ((#147,#148),.ROTATIONALMASSUNIT.,$);
#150= IFCDERIVEDUNITELEMENT (#21,1);
#151= IFCDERIVEDUNITELEMENT (#28,1);
#152= IFCDERIVEDUNITELEMENT (#43,-1);
   #153= IFCDERIVEDUNIT ((#150,#151,#152),.ROTATIONALSTIFFNESSUNIT.,$);
#154= IFCDERIVEDUNITELEMENT (#28,5);
   #155= IFCDERIVEDUNIT ((#154),.SECTIONAREAINTEGRALUNIT.,$);
#156= IFCDERIVEDUNITELEMENT (#28,3);
   #157= IFCDERIVEDUNIT ((#156),.SECTIONMODULUSUNIT.,$);
#158= IFCDERIVEDUNITELEMENT (#45,1);
   #159= IFCDERIVEDUNIT ((#158),.SHEARMODULUSUNIT.,$);
      #160= IFCUNITASSIGNMENT ((#20,#21,#28,#36,#43,#45,#56,#98,#102,
                                #105,#114,#120,#122,#141,#144,#149,
                                #153,#155,#157,#159));
```

Figura 4.16 – Modelo IFC - Unidades

```
#n = IFCSIUNIT ("Tipo de unidade","Prefixo","Nome");
#n = IFCMEASUREWITHUNIT ("Valor físico expresso em unidades especificas",
                         "Unidades em que o valor físico é expresso");
#n = IFCDIMENSIONALEXPONENTS ("Exponente de comprimento"," Exponente de massa"
                              ,"Exponente de tempo"," Exponente de corrente
                              elétrica"," Exponente de temperatura
                              termodinâmica","Exponente de matéria","Exponente
                              de intensidade luminosa");
#n = IFCCONVERSIONBASEDUNIT ("Exponentes dimensionais","Tipo de unidade",
                             "Nome","Factor de Conversão");
#n = IFCDERIVEDUNITELEMENT ("Unidade","Exponente");
#n = IFCDERIVEDUNIT ("Grupo de unidades que define a unidade derivada","Tipo
                     de unidade","Tipo definido pelo utilizador");
#n = IFCUNITASSIGNMENT ("Unidades");
```

Figura 4.17 – Aplicação - Unidades

Para se definir uma unidade que não faça parte do sistema SI, é necessário fazer a respectiva conversão através do *IfcConversionBaseUnit*. Tomando como exemplo a conversão de radianos para graus apresentada na Figura 4.18, primeiro faz-se a identificação da unidade a converter com a entidade *IfcSIUnit* (radianos), depois identifica-se a quantidade dessa unidade que corresponde ao valor unitário da unidade para a qual se pretende fazer a conversão (graus), especificando o tipo de medição que é necessário efectuar com a classe *IfcMeasureWithUnit* (medição angular → 0.01745329 rad ≈ 1º). De seguida definem-se os exponentes dimensionais com o *IfcDimensionalExponents* onde neste caso todos os exponentes são nulos pois o grau é uma medida adimensional. Por fim juntam-se todas as propriedades no *IfcConversionBasedUnit*, que faz a conversão das unidades com informação calculada a partir de outros atributos, usando expressões definidas.

```
#40= IFCSIUNIT (*,.PLANEANGLEUNIT.,$,.RADIAN.);
   #41= IFCMEASUREWITHUNIT (IFCPLANEANGLEMEASURE(0.0174532925199433),#40);
#42= IFCDIMENSIONALEXPONENTS (0,0,0,0,0,0,0);
      #43= IFCCONVERSIONBASEDUNIT (#42,.PLANEANGLEUNIT.,'DEGREE',#41);
```

Figura 4.18 – Exemplo – Conversão de unidades.

As unidades de grandezas fundamentais ficam a cargo da entidade *IfcSIUnit*, quanto às unidades de grandezas derivadas necessitam de pelo menos duas entidades para fazer a sua definição, uma ou mais entidades *IfcDerivedUnitElement* e uma *IfcDerivedUnit* que define a relação entre as unidades fundamentais e o tipo de unidade derivada em que resulta essa relação, como demonstrado na Figura 4.19.

O Pascal [Pa] é exemplo de uma unidade de grandeza derivada. É uma unidade padrão de pressão e tensão no sistema internacional de unidades (SI). Equivale à força de 1 Newton [N] aplicada uniformemente sobre uma superfície de 1 m², logo Pa = [N/m²]. Para se definir esta unidade derivada através do uso de entidades, primeiro identificam-se quais as instâncias que possuem as unidades fundamentais necessárias (#20 → m²; #21 → N) e seus exponentes no *IfcDerivedUnitElement*. O sinal positivo num exponente define a unidade em numerador ("1" → N) e o sinal negativo no exponente define a unidade em denominador ("-1" → m²). Para concluir a construção da unidade derivada utiliza-se o *IfcDerivedUnit* que faz a junção das unidades fundamentais, multiplicando as unidades que estão em denominador e multiplicando as unidades que estão em denominador.

```
#20= IFCSIUNIT (*,.AREAUNIT.,$,.SQUARE_METRE.);
#21= IFCSIUNIT (*,.FORCEUNIT.,$,.NEWTON.);
   #142= IFCDERIVEDUNITELEMENT (#21,1);
   #143= IFCDERIVEDUNITELEMENT (#20,-1);
      #144= IFCDERIVEDUNIT ((#142,#143),.PLANARFORCEUNIT.,$);
```

Figura 4.19 – Exemplo – Unidade derivada

A entidade que será atribuída ao *IfcProject* e que desse modo fará a atribuição das unidades a todo o ficheiro é a *IfcUnitAssignment*, é nela que se deve fazer o agrupamento das entidades de unidades. Portanto devem ser agrupadas nessa classe as unidades fundamentais no sistema SI (*IfcSIUnit)*, as unidades convertidas (*IfcConversionBasedUnit)* e as unidades derivadas (*IfcDerivedUnit*). No caso das unidades convertidas não se faz a atribuição da unidade base, pois resultaria num conflito de unidades, por exemplo, apenas se atribui a instância "#43" que define os graus, a instância "#40" que define os radianos não é atribuída.

4.3.3.5. DATA – Projecto

O *IfcProject* é uma das entidades fundamentais que deve sempre fazer parte de um qualquer ficheiro IFC, tem de ser incluído exactamente uma vez, servindo como raiz dos elementos num ficheiro de troca IFC para a visualização coordenada. O projecto estabelece o contexto para troca e partilha de informações e pode representar um projecto de construção, mas não necessariamente. Poderá indicar a realização de um projecto de engenharia, construção, design ou manutenção de actividades que conduzem a um produto. O projecto fornece um directório de tipos de objectos e modelos de propriedades contidos por meio de relações de declaração. Em suma, o propósito fundamental do *IfcProject* é fornecer a instância raiz e o contexto para todos os outros itens incluídos no ficheiro de troca, como demonstrado na Figura 4.20 e na Figura 4.21.

```
/* Projecto */
   #180= IFCPROJECT ('GUID',#5,'Project',$,$,$,$,(#202,#203),#160);
```

Figura 4.20 – Modelo IFC – O projecto

```
#n= IFCPROJECT ("GUID","Historial do usuário","Nome","Descrição","Tipo do
                Objecto","Nome completo","Fase","Contexto de representação",
                "Unidades em contexto");
```

Figura 4.21 – Aplicação – O projecto

Para facilitar a leitura dos exemplos expostos, os *Globally Unique Identifiers* do arquivo IFC, como por exemplo "0QjBRF7yLCTh4HRF3GOF1t", aparecem representados apenas pela inscrição "GUID". Não é demais referir que num ficheiro IFC o GUID deve ser necessariamente uma *string* de 22 caracteres, não podendo existir dois GUID's iguais no mesmo código, a existência de um GUID que não siga estas regras poderá implicar na não definição do objecto que contempla esse GUID. Um *IfcProject* tem um GUID como muitos elementos IFC, um *link* para o *IfcOwnerHistory*, referências para as unidades a usar no ficheiro e contexto da representação geométrica que permite a representação de formas e figuras nesse ficheiro.

4.3.3.6. DATA – Localização do Projecto

A localização do projecto é feita pela definição de um sistema de eixos global, que lhe será atribuído através do contexto de representação geométrica e desse modo essa sistema de eixos tornar-se-á válido para todo o modelo IFC (Figura 4.22 e Figura 4.23). Para além desse sistema de eixos global podem ainda coexistir outros sistemas de coordenadas locais, que por exemplo, poderão servir de auxílio para uma definição de elementos a nível local.

```
/* Localização do Projecto */
   #190= IFCAXIS2PLACEMENT2D (#8,#15);
   #191= IFCAXIS2PLACEMENT3D (#7,$,$);
      #192= IFCLOCALPLACEMENT ($,#191);
         #193= IFCBUILDING ('GUID',#5,$,$,$,#192,$,$,.ELEMENT.,$,$,#194);
   #194= IFCPOSTALADDRESS (.HOME.,$,$,'Arões','Rua da Aldeia','n°93','Vale de
                           Cambra','Aveiro','3730-001','Portugal');
         #195= IFCBUILDINGSTOREY ('GUID',#5,'Ground',$,$,#192,$,'Rés-do-Chão',
                                 .COMPLEX.,0.);
         #196= IFCBUILDINGSTOREY ('GUID',#5,'Floor',$,$,#192,$,'1°Piso',
                                 .COMPLEX.,3.);
   #197= IFCSITE ('GUID',#5,'Site',$,$,#192,$,$,.ELEMENT.,(40,47,57),(8,17,54)
                  634.,$,$);
            #198= IFCRELAGGREGATES ('GUID',#5,$,$,#197,#193);
            #199= IFCRELAGGREGATES ('GUID',#5,$,$,#193,(#195,#196));
```

Figura 4.22 – Modelo IFC – O projecto

```
#n= IFCAXIS2PLACEMENT2D ("Localização","Direção de referência");
#n= IFCAXIS2PLACEMENT3D ("Localização","Eixos","Direção de Referência");
#n= IFCLOCALPLACEMENT ("Posicionamento relacionado","Posicionamento
                       relativo");
#n= IFCBUILDING ("GUID","Historial do usuário","Nome","Descrição","Tipo do
                 objecto","Colocação do Objecto","Representação","Nome
                 completo","Tipo de composição","Elevação da altura de
                 referência","Elevação do terreno","Morada do Edifício");
```

```
#n= IFCBUILDINGSTOREY ("GUID","Historial do usuário","Nome","Descrição","Tipo
                       do objecto","Colocação do Objecto","Representação",
                       "Nome completo","Tipo de composição","Elevação");
#n= IFCPOSTALADDRESS ("Propósito","Descrição","Propósito definido pelo
                      utilizador","Localização interna","Morada","Caixa postal"
                      ,"Cidade","Região","Código Postal","País");
#n= IFCSITE (("GUID","Historial do usuário","Nome","Descrição","Tipo do
              objecto","Colocação do Objecto","Representação","Nome completo",
              "Tipo de composição","Latitude de referência","Longitude de
               referência","Elevação de referência","Designação do local no
               sistema regional","Morada do local");
#n= IFCRELAGGREGATES ("GUID","Historial do usuário","Nome","Descrição",
                      "Objectos a relacionar","Objectos relacionados");
```

Figura 4.23 – Aplicação – O projecto

O *IfcAxis2Placement2D* fornece a localização e orientação para que seja possível colocar objectos num espaço bidimensional, enquanto o *IfcAxis2Placement3D* faz essa colocação num espaço tridimensional. No caso do atributo "direcção de referência" não ser atribuído, são dadas direcções de referência automaticamente, segundo o eixo *x* [1.,0.] e segundo o eixo *y* [0.,1.] para o caso bidimensional e segundo o eixo *x* [1.,0.,0.], *y* [0.,1.,0.] e *z* [0.,0.,1.] para o caso tridimensional. Como os eixos são colocados segundo um principio de perpendicularidade e a direcção pode ser gerada por referência, tem-se então um conjunto de eixos cuja definição depende principalmente do ponto de aplicação. O sistema de eixos global definido neste grupo pela entidade *IfcAxis2Placement3D* será um atributo inverso do *IfcGeometricRepresentationContext* que por sua vez será um atributo inverso do *IfcProject*, é desta forma que o sistema de eixos definido ficará associado ao projecto.

Tal como o *IfcGridPlacement*, o *IfcLocalPlacement* é um subtipo do *IfcObjectPlacement*. O posicionamento de um produto no espaço pode ser relativamente ao sistema de coordenadas global, relativo ao posicionamento de um objecto ou produto, ou relativo aos eixos cartesianos. O *IfcLocalPlacement* permite fazer o posicionamento relativo de um produto em relação a um outro produto ou um posicionamento absoluto de um produto no contexto de representação geométrica do projecto.

O *IfcBuilding* pode conter informação espacial de elementos ou anotações e grelhas que lhe estejam directamente associados e pode também conter informações acerca da localização do edifício se for usado o *IfcPostalAdress* como atributo inverso. No caso dos elementos da construção, estes são associados ao *IfcBuilding* usando o *IfcRelContainedInSpatialStructure*. Se for pretendida uma atribuição dos elementos a um determinado nível específico, a associação deve ser feita com o mesmo seguimento lógico, mas desta vez associando os elementos ao *IfcBuildingStorey*. O *IfcSite* é a definição de uma área de terreno (sítio) onde se dará lugar à construção de um edifício. Um sítio identificado deste modo pode incluir pode incluir uma definição de ponto de referência geográfica (posição global usando o WGS84 com longitude, latitude e elevação), no caso do exemplo sabe-se que o edifício está localizado a 40º 47' 57" N e 8º 17' 54" O a uma altitude de 634 metros. Para permitir a composição e relações de agregação entre classes, tem-se o *IfcRelAggregates* que neste caso é a entidade responsável por estabelecer o contexto das ligações entre as diversas entidades de contenção espacial de elementos. O *IfcBuildingStorey* deve ser colocado em relação ao posicionamento do *IfcBuilding* e este por sua vez deve ser colocado em relação ao posicionamento do *IfcSite*.

A Figura 4.24 demonstra o encadeamento e relação que deve ser feito entre as entidades mencionadas, de modo a fazer um correcto arranjo espacial dos elementos.

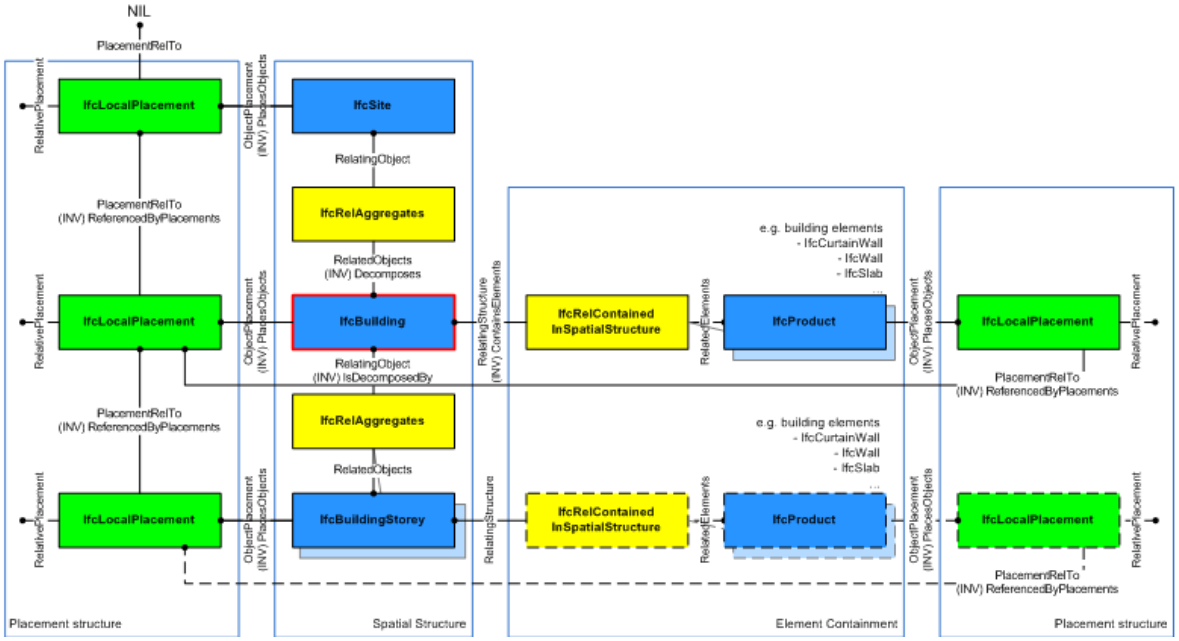


Figura 4.24 – Composição do Edifício (BuildingSMART 2013)

Em suma, o *IfcSite* permite conter elementos que estejam colocados no terreno, isto é, fora do edifício. O *IfcBuildingStorey* permite a definição espacial dos elementos por níveis e o *IfcBuilding* é uma entidade mais geral que por predefinição contém os elementos no espaço, mas em nenhum nível específico.

4.3.3.7. DATA – Representação Geométrica

O contexto em que se aplicam as várias representações de forma dos produtos dentro de um projecto é dado pela entidade *IfcGeometricRepresentationContext* e portanto, todos os ficheiros IFC que contenham geometria devem definir um contexto de representação geométrica (Figura 4.25 e Figura 4.26). Para além da forma, o *IfcGeometricRepresentationContext* pode também definir a precisão numérica aplicável aos elementos e a representação geométrica definida nesse contexto. As representações geométricas devem ser atribuídas ao *IfcProject* por atributo inverso. É ainda através desta instância que se definem os sistemas de coordenadas e de eixos global, o principal contexto de representação pode também fornecer a direcção do Norte verdadeiro. A linha da direcção Norte é dada pela união de dois pontos, o ponto origem do referencial e um outro ponto definido pelo utilizador de modo a formar um ângulo com o eixo dos y's que corresponde à direcção e sentido pré-definido.

Existem alguns identificadores para representação de formas que incluem valores pré-definidos, podendo ser encontrados no site da BuildingSMART, mais especificamente em:

http://www.buildingsmart-tech.org/ifc/IFC4/final/html/schema/ifcrepresentationresource/lexical/ifcshaperepresentation.htm

```
/* Representação geométrica */
   #202= IFCGEOMETRICREPRESENTATIONCONTEXT ('3D','Model',3,0.00001,#191,$);
   #203= IFCGEOMETRICREPRESENTATIONCONTEXT ('2D','Plan',2,0.00001,#191,$);
      #204= IFCGEOMETRICREPRESENTATIONSUBCONTEXT ('Body','Model',#202,$,
                                                  .MODEL_VIEW.,$);
```

```
#205= IFCGEOMETRICREPRESENTATIONSUBCONTEXT ('Annotation','Plan',#203,$,
                                            .PLAN_VIEW.,$);
```

Figura 4.25 – Modelo IFC – Representação Geométrica

```
#n= IFCGEOMETRICREPRESENTATIONCONTEXT ("Identificador de contexto","Tipo de
                                       contexto","Dimensão espacial","Precisão"
                                       ,"Sistema coordenadas global","Direcção
                                       Norte");
#n= IFCGEOMETRICREPRESENTATIONSUBCONTEXT ("Identificador de contexto","Tipo de
                                          contexto","Contexto análogo",
                                         "Escala","Visualização","Visualização
                                          definida pelo utilizador");
```

Figura 4.26 – Aplicação – Representação Geométrica

É obrigatório o uso de uma instância do *IfcGeometricRepresentationContext* para a representação espacial do modelo (3D), porém, o uso de uma instância deste tipo para a representação plana (2D) é de carácter opcional. Esses são os principais contextos de representação geométrica, podendo ainda ser mais refinados usando subcontextos de informação geométrica recorrendo ao *IfcGeometricRepresentationSubContext*. Esta entidade é usada para definir tipos de representação semanticamente distintos para diferentes conteúdos de informação, dependendo da vista da representação e da escala de destino. Pode ser usado para controlar o nível de detalhe da representação da forma que assim é mais aplicável a este contexto de representação.

4.3.3.8. DATA – Elementos Estruturais

Até este ponto foram definidas as classes com carácter mais geral, que podem fazer parte integrante da maioria dos ficheiros IFC. As classes expostas na Figura 4.27 e Figura 4.28 são as que permitem a construção dos elementos propriamente ditos. Na linha de raciocínio usada o primeiro passo foi definir uma malha de pontos (nós) e respectivas condições de fronteira (deslocamentos e restrições de mobilidade), onde posteriormente será feita a conexão dos membros estruturais (barras) dos pilares e vigas e também das lajes. Os nós foram identificados como inferior e superior para distinguir os diferentes níveis e também foram numerados para tornar mais perceptível a ligação dos pilares, vigas e laje nos pontos respectivos.

```
/* Pilar 1 - Nó inferior */
   #220= IFCCARTESIANPOINT ((0.,0.,0.));
      #221= IFCVERTEXPOINT (#220);
         #222= IFCTOPOLOGYREPRESENTATION (#202,'Reference','Vertex',(#221));
            #223= IFCPRODUCTDEFINITIONSHAPE ($,$,(#222));
               #224= IFCSTRUCTURALPOINTCONNECTION ('GUID',#5,'Point Connection
                                                   1',$,$,$,#223,#225,$);

/* Pilar 1 - Restrições nó inferior - Fixo */
   #225= IFCBOUNDARYNODECONDITION ('Fixed',IFCBOOLEAN(.T.),IFCBOOLEAN(.T.),
                                   IFCBOOLEAN(.T.),IFCBOOLEAN(.T.),
                                   IFCBOOLEAN(.T.),IFCBOOLEAN(.T.));

/* Pilar 1 - Nó superior */
   #226= IFCCARTESIANPOINT ((0.,0.,3.));
      #227= IFCVERTEXPOINT (#226);
         #228= IFCTOPOLOGYREPRESENTATION (#202,'Reference','Vertex',(#227));
            #229= IFCPRODUCTDEFINITIONSHAPE ($,$,(#228));
```

```
                #230= IFCSTRUCTURALPOINTCONNECTION ('GUID',#5,'Point Connection
                                                    2',$,$,$,#229,#231,$);

/* Pilar 1 - Restrições nó superior - Livre */
   #231= IFCBOUNDARYNODECONDITION ('Free',IFCBOOLEAN(.F.),IFCBOOLEAN(.F.),
                                   IFCBOOLEAN(.F.),IFCBOOLEAN(.F.),
                                   IFCBOOLEAN(.F.),IFCBOOLEAN(.F.));
```

Figura 4.27 – Modelo IFC – Nós de ligação do Pilar 1

```
#n= IFCVERTEXPOINT ("Vértice");
#n= IFCTOPOLOGYREPRESENTATION ("Contexto dos itens","Identificador de
                               representação","Tipo de representação","Itens");
#n= IFCPRODUCTDEFINITIONSHAPE ("Nome","Descrição","Representação");
#n= IFCSTRUCTURALPOINTCONNECTION ("GUID","Historial do usuário","Nome",
                                  "Descrição","Tipo do objecto","Posicionamento
                                  do objecto","Representação","Condições
                                  aplicadas","Sistema de coordenadas");

#n= IFCBOUNDARYNODECONDITION ("Nome","Rigidez translacional em X","Rigidez
                              translacional em Y","Rigidez translacional em Z",
                             "Rigidez rotacional em X","Rigidez rotacional em Y"
                            ,"Rigidez rotacional em Z");
```

Figura 4.28 – Aplicação – Nós de ligação do Pilar 1

A definição de um nó é feita através do *IfcVertexPoint* que aplica esse nó ao ponto cartesiano que lhe está inversamente atribuído. A definição de nó estrutural ou ponto de suporte é dada pelo *IfcStructuralPointConnection*. Para que se possa fazer uma confirmação visual da correcta definição do nó, deve-se atribuir uma representação topológica ao elemento estrutural recorrendo à entidade *IfcTopologyRepresentation* cuja representação será feita através do *IfcProductDefinitionShape*. As referências topológicas variam consoante o tipo de elemento. Para a definição de um nó devem ser usados o identificador e tipo de representação "*Reference*" e "*Vertex*", para a definição de uma barra "*Reference*" e "*Edge*" e para a definição de um plano "*Reference*" e "*Face*".

O *IfcBoundaryNodeCondition* descreve as condições de apoio e conexões, permitindo restringir o movimento numa dada direcção através de valores booleanos de verdadeiro ou falso. No exemplo dado (Figura 4.29, Figura 4.30 e Figura 4.31) o nó inferior é encastrado e por isso todos os booleanos são identificados como verdadeiros, isto é, o movimento é restringido em todas as direcções. Note-se que o encastramento do nó inferior faz neste caso a representação analítica de uma sapata. Na definição das condições de fronteira dos nós superiores não seria necessário definir o *IfcBoundaryNodeCondition*, pois neste caso esta instância não vem atribuir nenhuma limitação já que todos os valores booleanos são identificados como falso, não definindo assim nenhum tipo de restrição ao movimento (livre). Neste caso particular esta referência foi feita por uma questão de coerência metodológica.

```
/* Pilar 1 - Membro estrutural (Objecto) */
   #300= IFCDIRECTION ((1.,0.,0.));
      #301= IFCSTRUCTURALCURVEMEMBER ('GUID',#5,'Curve Member 1A',$,$,$,#306,
                                      .RIGID_JOINED_MEMBER.,#300);
         #302= IFCRELCONNECTSSTRUCTURALMEMBER ('GUID',#5,$,$,#301,#224,$,$,
                                               $,$);
         #303= IFCRELCONNECTSSTRUCTURALMEMBER ('GUID',#5,$,$,#301,#230,$,$,
                                               $,$);

/* Pilar 1 - Membro estrutural (Representação) */
```

```
      #304= IFCEDGE (#221,#227);
         #305= IFCTOPOLOGYREPRESENTATION (#202,'Reference','Edge',(#304));
            #306= IFCPRODUCTDEFINITIONSHAPE ($,$,(#305));
```

Figura 4.29 – Modelo IFC – Membro estrutural do Pilar 1

```
/* Viga 1 - Membro estrutural (Objecto) */
      #350= IFCDIRECTION ((0.,0.,1.));
         #351= IFCSTRUCTURALCURVEMEMBER ('GUID',#5,'Curve Member 1B',$,$,$,#356,
                                         .RIGID_JOINED_MEMBER.,#350);
            #352= IFCRELCONNECTSSTRUCTURALMEMBER ('GUID',#5,$,$,#351,#230,$,$,
                                                  $,$);
            #353= IFCRELCONNECTSSTRUCTURALMEMBER ('GUID',#5,$,$,#351,#250,$,$,
                                                  $,$);

   /* Viga 1 - Membro estrutural (Representação) */
      #354= IFCEDGE (#227,#247);
         #355= IFCTOPOLOGYREPRESENTATION (#202,'Reference','Edge',(#354));
            #356= IFCPRODUCTDEFINITIONSHAPE ($,$,(#355));
```

Figura 4.30 – Modelo IFC – Membro estrutural da Viga 1

```
   #n= IFCSTRUCTURALCURVEMEMBER ("GUID","Historial do usuário","Nome","Descrição"
                                ,"Tipo do objecto","Posicionamento objecto",
                                "Representação","Tipo pré-definido","Eixos");
   #n= IFCRELCONNECTSSTRUCTURALMEMBER ("GUID","Historial do usuário","Nome",
                                      "Descrição","Membro estrutural relativo",
                                      "Conexão estrutural relacionada","Condições
                                      aplicadas","Condições adicionais",
                                      "Comprimento apoiado","Sistema de
                                      coordenadas");
   #n= IFCEDGE ("Inicio do traço","Fim do traço");
```

Figura 4.31 – Aplicação – Membro estrutural do Pilar 1 e/ou Viga 1

A definição estrutural dos pilares e vigas é feita da mesma forma e com uma metodologia semelhante à definição dos nós. Agora, em vez de usada a entidade *IfcVertexPoint* é utilizada a entidade *IfcEdge* que juntamente com o *IfcStructuralCurveMember* faz a definição estrutural de uma barra e no lugar do *IfcStructuralPointConnection* é usado o *IfcRelConnectsStructuralMember*. Note-se que o uso desta entidade foi feita em duplicado porque se pretende que as barras estejam unidas por dois pontos. Estruturalmente a diferença entre pilares e vigas está apenas na orientação da peça, uma viga tem o seu eixo baricêntrico contido num plano horizontal e um pilar tem o seu eixo baricêntrico num plano vertical. Tanto para a situação dos pilares como para a situação das vigas poderia ter sido utilizadas as classes *IfcDirection* definidas inicialmente, apenas se voltou a redefinir estas classes para facilitar a mudança de direcção de um dos pilares ou vigas, caso de se revelar necessário.

De seguida, a Figura 4.32 e a Figura 4.33 procuram fazer a demontração da definição estrutural de uma laje.

```
   /* Laje 1 - Membro estrutural (Objecto) */
      #390= IFCSTRUCTURALSURFACEMEMBER ('GUID',#5,'Surface Member
                                       1C',$,$,$,#400,.BENDING_ELEMENT.,0.22);
      #391= IFCSTRUCTURALSURFACECONNECTION ('GUID',#5,'Surface Connection 1',$,$
                                           ,$,#400,$,$);
         #392= IFCRELCONNECTSSTRUCTURALMEMBER ('GUID',#5,$,$,#390,#230,$,$,$,$);
         #393= IFCRELCONNECTSSTRUCTURALMEMBER ('GUID',#5,$,$,#390,#250,$,$,$,$);
         #394= IFCRELCONNECTSSTRUCTURALMEMBER ('GUID',#5,$,$,#390,#270,$,$,$,$);
         #395= IFCRELCONNECTSSTRUCTURALMEMBER ('GUID',#5,$,$,#390,#290,$,$,$,$);
```

```
/* Laje 1 - Membro estrutural (Representação) */
   #396= IFCPOLYLOOP ((#226,#246,#286,#266));
      #397= IFCFACEOUTERBOUND (#396,.T.);
         #398= IFCFACESURFACE ((#397),$,.T.);
            #399= IFCTOPOLOGYREPRESENTATION (#202,'Reference','Face',(#398));
               #400= IFCPRODUCTDEFINITIONSHAPE ($,$,(#399));
```

Figura 4.32 – Modelo IFC – Membro estrutural da Laje 1

```
#n= IFCSTRUCTURALSURFACEMEMBER ("GUID","Historial do usuário","Nome",
                               "Descrição","Tipo do objecto","Posicionamento
                               do objecto","Representação","Tipo pré-definido"
                               ,"Espessura");
#n= IFCSTRUCTURALSURFACECONNECTION ("GUID","Historial do usuário","Nome",
                                   "Descrição","Tipo do objecto",
                                   "Posicionamento do objecto","Representação"
                                   ,"Condições aplicadas");

#n= IFCPOLYLOOP ("Poligono");
#n= IFCFACEOUTERBOUND ("Limites","Orientação");
#n= IFCFACESURFACE ("Limites","Superfície","Sentido da normal");
```

Figura 4.33 – Aplicação – Membro estrutural da Laje 1

Alerta-se para a existência de erros na definição da laje do exemplo dado na Figura 4.32. Infelizmente, apesar de ter sido feito um longo estudo desta ocorrência em particular e das inúmeras alterações feitas num processo do tipo "tentativa-erro" a fim de se tornar a laje perfeitamente funcional, a concretização não foi alcançada. Infelizmente, devido ao escasso tempo disponível para esta investigação, software inexistente para o esquema IFC4 e inúmeras dificuldades sentidas, não foi possível identificar com exactidão qual o correcto modo de definição de uma laje estrutural. O melhor resultado alcançado foi a obtenção da representação da laje, o que significa que a definição estrutural da laje foi conseguida, mas à cota 0 metros e não à cota 3 metros como se pretendia. Foram definidos vários conjuntos de eixos na tentativa de resolução deste problema mas que acabaram sempre sem sucesso. Ao abrir com o programa "Constructivity" o modelo IFC4 criado, não é mostrada nenhuma janela com indicações e alertas de erros, o que leva a acreditar que o problema da não definição da laje possa estar na falta de alguma instância que não foi contemplada. Possivelmente para se resolver este problema será necessário um estudo de novas instâncias de entidades que possam contornar a questão e façam a definição dos elementos de outra forma. Espera-se que com estas indicações permitam que futuras investigações consigam alcançar os resultados pretendidos. Os inúmeros testes que foram realizados na tentativa de solucionar o problema não são expostos de forma propositada, pois não se pretende que sejam descartadas hipóteses numa futura análise do problema.

Para finalizar a determinação dos elementos estruturais é necessário estabelecer uma relação entre os vários elementos e o *IfcStructuralAnalysisModel* (classe que será exposta mais à frente em momento oportuno), relação essa que é feita através do *IfcRelAssignToGroup* que faz a atribuição de definições de objectos a um determinado grupo (Figura 4.34 e Figura 4.35).

```
/* Atribuição dos nós e membros estruturais ao modelo de análise estrutural */
   #401= IFCRELASSIGNSTOGROUP ('GUID',#5,$,$,(#224,#230,#301,#244,#250,#311,
                              #264,#270,#321,#284,#290,#331,#351,#361,#371,
                              #381,#390),.PRODUCT.,#875);
```

Figura 4.34 – Modelo IFC – Atribuição dos nós e membros estruturais ao modelo de análise estrutural

```
#n= IFCRELASSIGNSTOGROUP ("GUID","Historial do usuário","Nome","Descrição",
                          "Objectos relacionados","Tipo dos objectos
                          relacionados","Grupo relativo");
```

Figura 4.35 – Aplicação – Atribuição dos nós e membros estruturais ao modelo de análise estrutural

4.3.3.9. DATA – Elementos Arquitectónicos

Definidos os elementos de forma estrutural passa-se agora para a definição dos elementos mas de forma arquitectónica (Figura 4.36 e Figura 4.37). Esta definição arquitectónica está mais ligada ao conceito de geometria e das informações que se podem retirar a partir daí, por isso, foram ainda definidas, a título de exemplo, algumas propriedades da secção.

```
/* Pilares - Secção e características  */
   #425= IFCRECTANGLEPROFILEDEF (.AREA.,'20x30',$,0.20,0.30);
      #426= IFCEXTRUDEDAREASOLID(#425,#191,#13,3.0);
         #427= IFCSHAPEREPRESENTATION(#202,'Body','SweptSolid',(#426));
            #428= IFCPRESENTATIONLAYERASSIGNMENT('Columns Profile','No
                                                 description',(#427),$);
            #429= IFCPRODUCTDEFINITIONSHAPE($,$,(#427));
   #430= IFCPROPERTYSINGLEVALUE ('MassPerLength',$,IFCMASSPERLENGTHMEASURE(25)
                                ,$);
   #431= IFCPROPERTYSINGLEVALUE ('CrossSectionArea',$,IFCAREAMEASURE(0.06),$);
   #432= IFCPROPERTYSINGLEVALUE ('MomentOfInertiaY',$,
                                 IFCMOMENTOFINERTIAMEASURE(0.0002),$);
   #433= IFCPROPERTYSINGLEVALUE ('MomentOfInertiaZ',$,
                                IFCMOMENTOFINERTIAMEASURE(0.00045),$);
   #434= IFCPROPERTYSINGLEVALUE ('TorsionalSectionModulus',$,
                                IFCSECTIONMODULUSMEASURE(0.622),$);
      #435= IFCPROFILEPROPERTIES ('Pset_ProfileMechanical',$,(#430,#431,#432,
                                  #433,#434),#425);
   #436= IFCCARTESIANPOINT ((0.0,0.0,0.0));
      #437= IFCAXIS2PLACEMENT3D (#436,#13,#9);
         #438= IFCLOCALPLACEMENT (#192,#437);
               #439= IFCCOLUMN ('GUID',#5,'Column 1',$,$,#438,#429,$,
                                .COLUMN.);
                  #440= IFCRELCONNECTSSTRUCTURALELEMENT ('GUID',#5,$,$,(#301),
                                                         #439);
   #441= IFCCARTESIANPOINT ((5.5,0.,0.));
      #442= IFCAXIS2PLACEMENT3D (#441,#13,#9);
         #443= IFCLOCALPLACEMENT (#192,#442);
               #444= IFCCOLUMN ('GUID',#5,'Column 2',$,$,#443,#429,$,
                                .COLUMN.);
                  #445= IFCRELCONNECTSSTRUCTURALELEMENT ('GUID',#5,$,$,(#311),
                                                         #444);
   #446= IFCCARTESIANPOINT ((0.,5.5,0.));
      #447= IFCAXIS2PLACEMENT3D (#446,#13,#9);
         #448= IFCLOCALPLACEMENT (#192,#447);
               #449= IFCCOLUMN ('GUID',#5,'Column 3',$,$,#448,#429,$,
                                 .COLUMN.);
                  #450= IFCRELCONNECTSSTRUCTURALELEMENT ('GUID',#5,$,$,(#321),
                                                         #449);
   #451= IFCCARTESIANPOINT ((5.5,5.5,0.));
      #452= IFCAXIS2PLACEMENT3D (#451,#13,#9);
         #453= IFCLOCALPLACEMENT (#192,#452);
               #454= IFCCOLUMN ('GUID',#5,'Column 4',$,$,#453,#429,$,
                                .COLUMN.);
                  #455= IFCRELCONNECTSSTRUCTURALELEMENT ('GUID',#5,$,$,(#331),
                                                         #454);
```

```
#456= IFCRELCONTAINEDINSPATIALSTRUCTURE ('GUID',#5,$,$,(#439
                                         ,#444,#449,#454),
                                         #196);
```

Figura 4.36 – Modelo IFC – Definição da arquitectura dos pilares e algumas definições

```
#n= IFCRECTANGLEPROFILEDEF ("Tipo de perfil","Nome do perfil","Posição",
                            "Dimensão X","Dimensão Y");
#n= IFCEXTRUDEDAREASOLID ("Swept Área","Posição","Direção da extrusão",
                          "Profundidade");
#n= IFCSHAPEREPRESENTATION ("Contexto dos Itens","Identificador de
                            representação","Tipo de representação","Itens");
#n= IFCPRESENTATIONLAYERASSIGNMENT ("Nome","Descrição","Itens atribuídos",
                                    "Identificação");
#n= IFCPRODUCTDEFINITIONSHAPE ("Nome","Descrição","Representação");

#n= IFCPROPERTYSINGLEVALUE ("Nome","Descrição","Valor nominal","Unidades");
#n= IFCPROFILEPROPERTIES ("Nome","Descrição","Propriedades","Definição da
                          secção");

#n= IFCFOOTING ("GUID","Historial do usuário","Nome","Descrição","Tipo do
                objecto","Posicionamento do objecto","Representação","Etiqueta"
                ,"Tipo pré-definido");
#n= IFCCOLUMN ("GUID","Historial do usuário","Nome","Descrição","Tipo do
               objecto","Posicionamento do objecto","Representação","Etiqueta"
               ,"Tipo pré-definido");
#n= IFCBEAM ("GUID","Historial do usuário","Nome","Descrição","Tipo do
             objecto","Posicionamento do objecto","Representação","Etiqueta",
             "Tipo pré-definido");
#n= IFCSLAB ("GUID","Historial do usuário","Nome","Descrição","Tipo do
             objecto","Posicionamento do objecto","Representação","Etiqueta",
             "Tipo pré-definido");
#n= IFCRELCONNECTSSTRUCTURALELEMENT ("GUID","Historial do usuário","Nome",
                                     "Descrição","Conexão geométrica",
                                     "Elementos a relacionar","Elementos
                                     relacionados");
#n= IFCRELCONTAINEDINSPATIALSTRUCTURE ("GUID","Historial do usuário","Nome",
                                       "Descrição","Elementos relacionados",
                                       "Espaço a relacionar");
```

Figura 4.37 – Aplicação – Definição da arquitectura dos pilares e algumas definições

Como se pode ver pelos exemplos de aplicação, a metodologia de construção do código IFC dos elementos de construção como sapatas, pilares, vigas e laje é bastante semelhante. Já que todos têm secção rectangular todos foram criados a partir da extrusão de uma secção rectangular usando o *IfcExtrudedAreaSolid* com a entidade *IfcRectangleProfileDef* inversamente relacionada, mas repare-se que poderiam ser usadas outras secções de entidades subtipo da *IfcProfileDef* ou até mesmo a criação de secções irregulares pela definição do seu contorno através do *IfcPolyLoop*. Depois, de modo a fazer a definição do tipo de elementos estrutural que dado objecto representa (se se trata de um pilar, viga, laje, etc.) usa-se uma instância IFC respectiva para esse efeito (*IfcColumn*, *IfcBeam*, *IfcSlab*, etc.).

A entidade *IfcRelConnectsStructuralElement* faz uma associação de conectividade entre elementos numa relação de um para um. O conceito de dois elementos serem associados fisicamente ou logicamente é descrito de forma independente pela conectividade entre elementos. Essa ligação pode ser feita em termos de representação da secção, definição dos objectos ou atribuição de geometria.

O *IfcRelContainedInSpatialStructure* é usado para fazer a atribuição de elementos a um determinado nível espacial da estrutura do projecto, sendo que cada objecto apenas pode ser aplicado uma única vez a um dado nível dessa estrutura.

A figura 4.38 ilustra o funcionamento da hierarquia para a agregação dos elementos estruturais ao *IfcBuilding*. Como se pode observar existem elementos (paredes no exemplo) que podem ser directamente atribuídos a um determinado nível, mas existem também elementos que poderão não pertencer a um determinado nível específico (escadas no exemplo) e por isso a atribuição não é feita a um nível ou piso, mas sim ao próprio edifício.

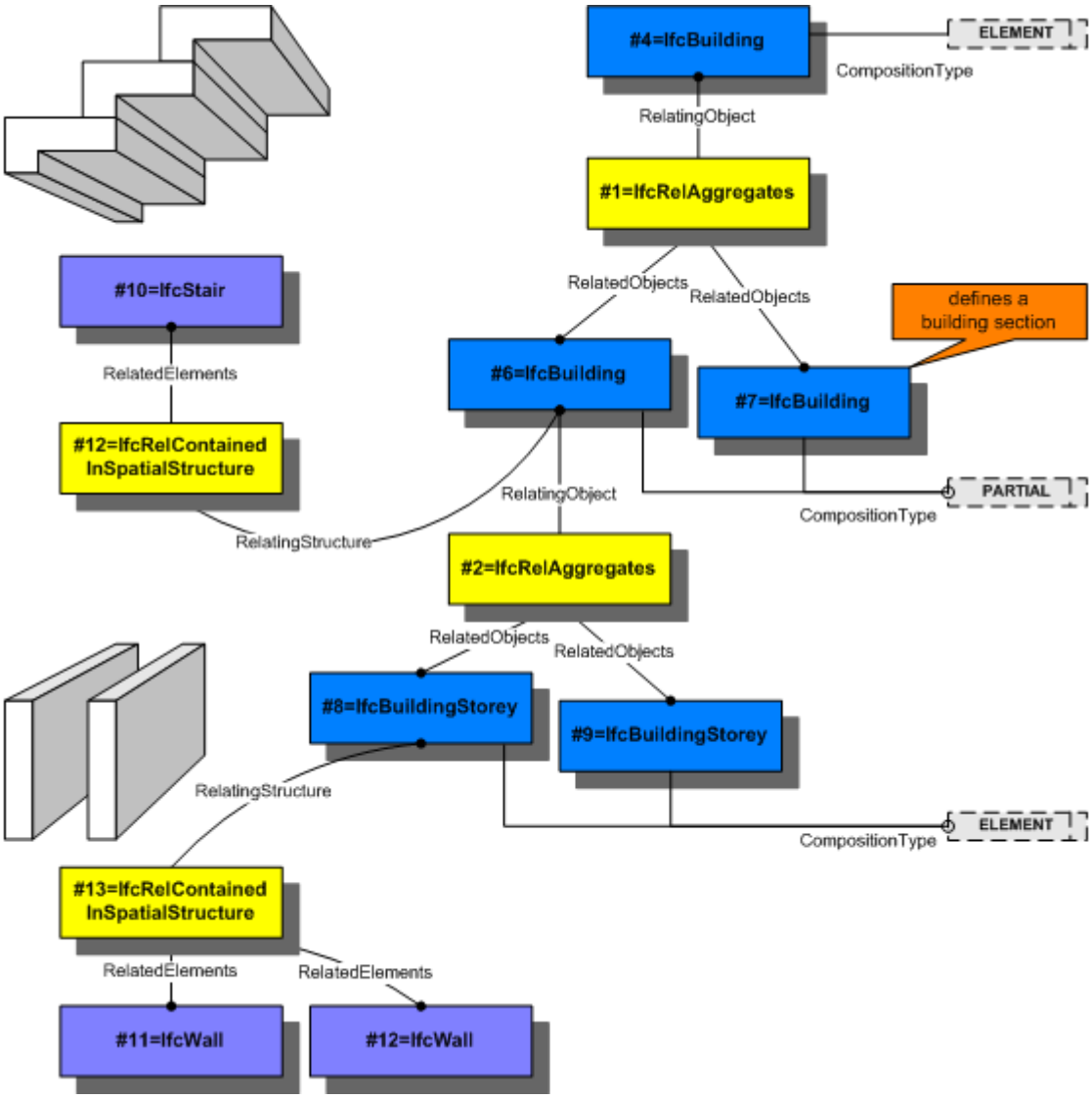


Figura 4.38 – Relação para a contenção espacial de elementos (BuildingSMART 2013)

Por último importa ainda distinguir os elementos superficiais de elementos sólidos. Os elementos superficiais baseiam-se na modelação de faces que possibilita a criação de formas livres e objectos com formatos complexos que também podem ser usados nas ferramentas BIM para a construção de objectos geométricos com faces planas ou multifacetadas. Os elementos tridimensionais são criados

pela definição dos limites fechados que compreendem um determinado volume. Esta técnica preocupa-se sobretudo com a representação geométrica e não tanto com as propriedades físicas do elemento, sendo este tipo de informações frequentemente denominados de "*dummy data*". Os elementos sólidos são mais adequados para os modelos BIM pois permitem a indexação de informações aos objectos (*smart objects*). As duas técnicas de representação mais conhecidas usadas na modelação de elementos são a CSG e a BREP, a primeira faz a representação de sólidos a partir da extrusão de uma secção ou região numa dada direcção e a segunda faz a representação de um sólido através da conexão de vértices e arestas que representam os limites desse sólido. (Dimyadi, Spearpoint, and Amor 2007)

4.3.3.10. DATA – Materiais e Propriedades

Agora que os elementos já se encontram analiticamente e geometricamente definidos é possível completar essa definição pela atribuição de propriedades físicas a esses mesmos elementos (Figura 4.39 e Figura 4.40). Na alínea anterior já foram atribuídos algumas propriedades relacionadas com a secção do objecto, agora as propriedades definidas estão relacionadas com o material que será atribuído aos elementos estruturais. Pretende-se que os elementos sejam em Betão C20/25 e para isso definiram-se algumas propriedades desse material, mas poderiam ter sido definidas ainda mais propriedades do material usando o mesmo método que foi usado no exemplo dado.

```
/* Pilares - Propriedades do material */
   #700= IFCMATERIAL ('C20/25',$,'Concrete');
   #701= IFCPROPERTYSINGLEVALUE ('MassDensity',$,
                                  IFCMASSDENSITYMEASURE(2407.31),$);
      #702= IFCMATERIALPROPERTIES ('Pset_MaterialCommon',$,(#701),#700);
   #703= IFCPROPERTYSINGLEVALUE ('YoungModulus',$,
                                IFCMODULUSOFELASTICITYMEASURE(23250000000),$);
   #704= IFCPROPERTYSINGLEVALUE ('ShearModulus',$,
                                 IFCMODULUSOFELASTICITYMEASURE(9964000000),$);
      #705= IFCMATERIALPROPERTIES ('Pset_MaterialMechanical',$,(#703,#704),
                                   #700);
   #706= IFCPROPERTYSINGLEVALUE ('ConstructionMethod',$,'In-Situ',$);
   #707= IFCPROPERTYSINGLEVALUE ('StructuralClass',$,'S1',$);
   #708= IFCPROPERTYSINGLEVALUE ('StrengthClass',$,'C20/25',$);
   #709= IFCPROPERTYSINGLEVALUE ('ExposureClass',$,'XC4',$);
   #710= IFCPROPERTYSINGLEVALUE ('ReinforcementVolumeRatio',$,
                                 IFCMASSDENSITYMEASURE(280),$);
   #711= IFCPROPERTYSINGLEVALUE ('ConstructionToleranceClass',$,'cm',$);
   #712= IFCPROPERTYSINGLEVALUE ('ConcreteCover',$,
                                 IFCPOSITIVELENGHTMEASURE(0.025),$);
   #713= IFCPROPERTYSINGLEVALUE ('ReinforcementStrengthClass',$,'S500',$);
      #714= IFCPROPERTYSET ('GUID',#5,'Pset_ConcreteElementGeneral',$,(#706,
                            #707,#708,#709,#710,#711,#712,#713));
         #715= IFCRELDEFINESBYPROPERTIES ('GUID',#5,$,$,(#301,#311,#321,#331,
                                          #439,#444,#449,#454),#714);
      #716= IFCMATERIALPROFILE ($,$,#700,#425,$,$);
         #717= IFCMATERIALPROFILESET ($,$,(#716),$);
            #718= IFCMATERIALPROFILESETUSAGE (#717,$,$);
               #719= IFCRELASSOCIATESMATERIAL ('GUID',#5,$,$,(#301,#311,#321,
                                               #331,#439,#444,#449,#454),#718);
```

Figura 4.39 – Modelo IFC – Propriedades atribuídas ao Pilar 1

```
#n= IFCMATERIAL ("Nome","Descrição","Categoria");
#n= IFCPROPERTYSINGLEVALUE ("Nome","Descrição","Valor nominal","Unidades");
#n= IFCMATERIALPROPERTIES ("Nome","Descrição","Propriedades","Material");
#n= IFCPROPERTYSET ("GUID","Historial do usuário","Nome","Descrição","Tipo
                    definido","Definido por","Ocorrência definida");
```

```
#n= IFCRELDEFINESBYPROPERTIES ("GUID","Historial do usuário","Nome",
                               "Descrição","Objectos relacionados","Definições
                               de propriedades a relacionar");
#n= IFCMATERIALPROFILE ("Nome","Descrição","Material","Secção","Prioridade",
                        "Categoria");
#n= IFCMATERIALPROFILESET ("Nome","Descrição","Secção do material","Secção
                           composta");
#n= IFCMATERIALPROFILESETUSAGE ("Definições da secção","Ponto de referência",
                                "Extensão da referência");
#n= IFCRELASSOCIATESMATERIAL ("GUID","Historial do usuário","Nome",
                              "Descrição","Objectos a relacionar","Material
                              relacionado");
```

Figura 4.40 – Aplicação – Propriedades atribuídas ao Pilar 1

A definição das várias características do material é então feita usando o *IfcPropertySingleValue*, que define uma propriedade por instância. As propriedades de natureza semelhante são depois englobadas no *IfcMaterialProperties* que faz a definição dessas propriedades singulares segundo uma determinada "*PropertySet*" pré-definida e por atributo inverso faz se a atribuição dessas propriedades ao material representado globalmente pelo *IfcMaterial*. Só assim se garante que o material mencionado nesta entidade contém de facto características físicas e não apenas o nome de um material que poderia ou não ser reconhecido pelo programa destino. A notação "*Pset_MaterialMechanical*" indica que as propriedades do conjunto de definições identificado faz parte da especificação IFC, devendo ser associada ao perfil do material com a entidade *IfcMaterialProfile*, que pode ser atribuído a uma secção de um dado objecto ou conjunto de objectos utilizando-se o *IfcRelAssociatesMaterial*.

Para além das propriedades das secções e materiais poderiam ser definidas muitas outras propriedades do tipo quantitativo ou alfanumérico, num encadeamento semelhante ao anterior, mas agora usando o *IfcPropertySet* para agrupar as várias entidades *IfcPropertySingleValue* e o *IfcRelDefinesByProperties* para fazer a associação dessas propriedades directamente a um objecto ou grupo de objectos.

4.3.3.11. DATA – Modelo de Análise Estrutural

Concluída toda a definição dos elementos estruturais podem agora ser definidas um conjunto de acções. Para que isso seja possível é preciso criar um modelo de análise estrutural através do *IfcStructuralAnalysisModel* como exemplificado na Figura 4.41 e na Figura 4.42. Assim sendo, o *IfcStructuralAnalysisModel* é a entidade que reúne os vários casos de carga aplicados à estrutura e as posiciona segundo um referencial que corresponde ao sistema de eixos global.

```
/* Modelo de análise estrutural */
   #875= IFCSTRUCTURALANALYSISMODEL ('GUID',#5,'Structural Analysis 1',$,$,
                                     .NOTDEFINED.,#191,(#900,#910,#920,#930),$,
                                     #192);
      #876= IFCRELDECLARES ('GUID',#5,'GROUP',$,#180,(#875));
```

Figura 4.41 – Modelo IFC – Modelo de Análise Estrutural

```
#n= IFCSTRUCTURALANALYSISMODEL ("GUID","Historial do usuário","Nome",
                                "Descrição","Tipo do objecto","Tipo pré-
                                definido","Orientação do plano","Casos de
                                carga","Resultados","Posicionamento
                                partilhado");
#n= IFCRELDECLARES ("GUID","Historial do usuário","Nome","Descrição","Contexto
                    relacionado","Definições");
```

Figura 4.42 – Aplicação – Modelo de Análise Estrutural

Pelo *IfcRelDeclares* é feita a declaração dos objectos ou propriedades ao *IfcProject*, neste caso fazendo também o seu agrupamento através do atributo de enumeração ".GROUP.". Ao ser feita esta relação e agrupamento das classes que definem objectos, torna-se possível a atribuição de características a todas as classes que fazem parte desse grupo, bastando agora apenas ser mencionado o grupo por atributo inverso em vez de relacionar individualmente cada uma das entidades.

4.3.3.12. DATA – Cargas Estruturais

Neste último conjunto de entidades são definidas as cargas a aplicar ao modelo de análise estrutural (Figura 4.43 e Figura 4.44). As forças definidas neste modelo IFC são apenas casos de carga usados para fazer uma explicação prática do modo de aplicação dos diferentes casos de carga usando entidades IFC, não contemplando nenhum tipo de combinação. O exemplo dado contempla cargas pontuais, lineares e uniformemente distribuídas num plano.

```
/* Pilar 1 - Carga linear */
   #900= IFCSTRUCTURALLOADCASE ('GUID',#5,'Structural Load Case 1',$,$,
                               .LOAD_CASE.,.NOTDEFINED.,.NOTDEFINED.,1.,$,
                               (0.,0.,0.));
   #901= IFCSTRUCTURALLOADLINEARFORCE ('Nominal',$,$,-1.64,$,$,$);
   #902= IFCSTRUCTURALLOADLINEARFORCE ('Nominal',$,$,-1.64,$,$,$);
      #903= IFCSTRUCTURALLOADCONFIGURATION ($,(#901,#902),((0.),(3.)));
         #904= IFCSTRUCTURALCURVEACTION ('GUID',#5,'Structural Curve Action 1'
                                       ,$,$,$,$,#903,.GLOBAL_COORDS.,.F.,$,
                                       .LINEAR.);
            #905= IFCRELCONNECTSSTRUCTURALACTIVITY ('GUID',#5,$,$,#301,#904);
            #906= IFCRELASSIGNSTOGROUP ('GUID',#5,$,$,(#904),.PRODUCT.,#900);

/* Pilar 3 - Carga pontual */
   #910= IFCSTRUCTURALLOADCASE ('GUID',#5,'Structural Load Case 2',$,$,
                               .LOAD_CASE.,.NOTDEFINED.,.NOTDEFINED.,1.,$,
                               (0.,0.,0.));
   #911= IFCSTRUCTURALLOADSINGLEFORCE ('Força 1',$,$,-2.54,$,$,$);
      #912= IFCSTRUCTURALLOADCONFIGURATION ($,(#911),((3.)));
         #913= IFCSTRUCTURALPOINTACTION ('GUID',#5,'Structural Point Action 1'
                                        ,$,$,$,$,#912,.GLOBAL_COORDS.,.F.);
            #914= IFCRELCONNECTSSTRUCTURALACTIVITY ('GUID',#5,$,$,#321,#913);
            #915= IFCRELASSIGNSTOGROUP ('GUID',#5,$,$,(#913),.PRODUCT.,#910);

/* Viga 1 - Carga trapezoidal */
   #920= IFCSTRUCTURALLOADCASE ('GUID',#5,'Structural Load Case 3',$,$,
                              .LOAD_CASE.,.NOTDEFINED.,.NOTDEFINED.,1.,$,
                              (0.,0.,0.));
   #921= IFCSTRUCTURALLOADLINEARFORCE ('Nominal',5,$,$,$,$,$);
   #922= IFCSTRUCTURALLOADLINEARFORCE ('Nominal',2,$,$,$,$,$);
      #923= IFCSTRUCTURALLOADCONFIGURATION ($,(#921,#922),((0.),(5.5)));
         #924= IFCSTRUCTURALCURVEACTION ('GUID',#5,'Structural Curve Action 2'
                                       ,$,$,$,$,#923,.GLOBAL_COORDS.,.F.,$,
                                       .LINEAR.);
            #925= IFCRELCONNECTSSTRUCTURALACTIVITY ('GUID',#5,$,$,#351,#924);
            #926= IFCRELASSIGNSTOGROUP ('GUID',#5,$,$,(#924),.PRODUCT.,#920);

/* Laje 1 - Carga uniformemente distribuida */
   #930= IFCSTRUCTURALLOADCASE ('GUID',#5,'Structural Load Case 4',$,$,
                               .LOAD_CASE.,.NOTDEFINED.,.NOTDEFINED.,1.,$,
                               (0.,0.,0.));
   #931= IFCSTRUCTURALLOADPLANARFORCE ('Nominal',$,$,-4);
      #932= IFCSTRUCTURALLOADCONFIGURATION ($,(#931),((0.)));
         #933= IFCSTRUCTURALPLANARACTION ('GUID',#5,'Structural Planar Action
                                         1',$,$,$,$,#932,.GLOBAL_COORDS.,
                                         .F.,$,.CONST.);
```

```
                #934= IFCRELCONNECTSSTRUCTURALACTIVITY ('GUID',#5,$,$,#390,#933);
                #935= IFCRELASSIGNSTOGROUP ('GUID',#5,$,$,(#933),.PRODUCT.,#930);

ENDSEC;

END-ISO-10303-21;
```

Figura 4.43 – Modelo IFC – Cargas estruturais

```
#n= IFCSTRUCTURALLOADCASE ("GUID","Historial do usuário","Nome","Descrição",
                           "Tipo do objecto","Tipo pré-definido","Tipo da
                           acção","Origem da acção","Coeficiente","Propósito",
                           "Coeficiente de peso próprio");
#n= IFCSTRUCTURALLOADLINEARFORCE ("Nome","Força linear em X","Força linear em
                                  Y","Força linear em Z","Momento linear em X",
                                  "Momento linear em Y","Momento linear em Z");
#n= IFCSTRUCTURALLOADCONFIGURATION ("Nome","Valores","Localização");
#n= IFCSTRUCTURALCURVEACTION ("GUID","Historial do usuário","Nome","Descrição"
                              ,"Tipo do objecto","Posicionamento do objecto",
                              "Representação","Carga aplicada","Referêncial",
                              "Projecção da carga","Tipo pré-definido");
#n= IFCRELCONNECTSSTRUCTURALACTIVITY ("GUID","Historial do usuário","Nome",
                                      "Descrição","Elementos a relacionar",
                                      "Actividade estrutural relacionada");

#n= IFCSTRUCTURALLOADSINGLEFORCE ("Força em X","Força em Y","Força em Z",
                                  "Momento em X","Momento em Y","Momento em Z");
#n= IFCSTRUCTURALPOINTACTION ("GUID","Historial do usuário","Nome","Descrição"
                              ,"Tipo do objecto","Posicionamento do objecto",
                              "Representação","Carga aplicada","Referêncial",
                              "Carga destabilizadora");

#n= IFCSTRUCTURALLOADPLANARFORCE ("Nome","Força plana em X","Força plana em
                                  Y","Força plana em Z");
#n= IFCSTRUCTURALPLANARACTION ("GUID","Historial do usuário","Nome",
                               "Descrição","Tipo do objecto","Posicionamento
                               do objecto","Representação","Carga aplicada",
                               "Referêncial","Projecção da carga","Tipo pré-
                               definido");
```

Figura 4.44 – Aplicação – Cargas estruturais

Na definição dos diferentes tipos de carga usa-se também uma metodologia que é bastante semelhante nos vários casos, todos são definidos com a entidade *IfcStructuralLoadCase*, mas existem alguns pormenores fundamentais sem os quais a definição não ficaria completa e por isso não seria funcional.

A acção de uma carga pontual é feita identificando as direcções que compõem essa carga e respectivos valores com instâncias da entidade *IfcStructuralLoadSingleForce*. Para caracterizar a acção pontual como uma acção que actua num ponto usa-se o *IfcStructuralPointAction*. Posto isso, faz-se a selecção da barra ou ponto de aplicação da acção por atributo inverso na entidade *IfcRelConnectsStructuralActivity*, que faz a relação entre uma determinada carga estrutural e o elemento, membro estrutural ou ligação estrutural onde essa carga está aplicada. No caso de a força estar aplicada numa barra, o referencial usado no *IfcStructuralLoadConfiguration* é um referencial local, quer isto dizer que, por exemplo, o valor "(3.)" do atributo "Localização" desta entidade refere-se ao ponto de extremidade de uma barra com 3 metros. Para reforçar esta demonstração suponhamos que o valor do atributo "Localização" era "(1,5)", isto significaria que a carga estava aplicada na barra a meio vão. Posto isto, a definição de uma carga linear é feita nos mesmos moldes, apenas variam as

classes que define o tipo e aplicação da carga. No caso do tipo de uma carga linear a classe a usar é a *IfcStructuralCurveAction*, classe supertipo da *IfcStructuralLinearAction* e no caso da aplicação a classe a usar é agora a *IfcLoadLinearForce*. A aplicação da carga linear às barras é também feita num referencial local, mas desta feita é necessário definir os dois pontos de aplicação que definem a carga linear (inicial e final) e os valores inicial e final da carga linear, desse modo é possível definir cargas lineares uniformemente distribuídas (se os valores das forças inicial e final foram iguais), cargas lineares de distribuição triangular (se um dos valor da força inicial ou final for nulo) e também cargas lineares trapezoidais (se os valores das forças inicial e final foram diferentes). Para a aplicação de cargas uniformemente distribuídas em lajes ou num plano, as entidades que diferem das utilizadas nos métodos anteriores são a *IfcStructuralLoadPlanarForce* e a *IfcStructuralPlanarAction*, todo o restante processo é semelhante.

Por fim, para se conclui a secção DATA introduz-se a *string* "ENDSEC;" e para dar como terminada toda a estrutura de troca IFC introduz-se a *string* "END-ISO-10303-21;".

4.4. Considerações Complementares

É importante reter que um qualquer ficheiro IFC é sempre definido segundo um determinado esquema, quer esteja sobre a forma STEP-file ou representação XML. O esquema faz a atribuição de significado aos nomes e relações das entidades dentre de um ficheiro IFC, sendo estático para todos os ficheiros IFC4 e publicado pela BuildingSMART segundo o esquema EXPRESS e também em formato XML de forma alternativa.

Os *Model View Definitions* (MVD) aparecem no topo do esquema como a parte do esquema que é descrita dentro de um determinado ficheiro IFC, que posteriormente descreve cada instância de um determinado componente de construção contido nesse ficheiro.

As instâncias contêm dados semânticos como o nome, a descrição, a geometria, o tipo, as relações entre outras instâncias, o posicionamento e relação espacial e as propriedades. Dentro da secção de atributos de entidades que definem tipos de elementos estruturais (por exemplo *IfcColumn*) é agora possível no esquema IFC4 definir tipos pré-definidos como um dos atributos (e através de enumerações), que dão indicações do uso e função desses elementos. As informações não geométricas podem ser anexadas às instâncias e também aos tipos, podendo várias instâncias referir o mesmo tipo.

O posicionamento dos objectivos pode ser absoluto ou relativo ao posicionamento de outro objecto. As instâncias que definem geometria visível podem ser reutilizadas e atribuídas a outras entidades. Salienta-se que nem toda a geometria definida ser necessariamente visível. Cada instância pode ainda ter mais do que uma representação geométrica e uma representação tridimensional ter mais do que uma cor atribuída.

Por último chama-se a atenção para o facto do esquema IFC4 ser repleto de relações, onde muitas delas são relações inversas definidas pelo próprio esquema e por isso não são directamente visíveis no texto de código (BuildingSMART 2013).

5

PROPOSTA DE IMPLEMENTAÇÃO DE CLASSES IFC NA CRIAÇÃO DE SOFTWARES

5.1. ENUNCIADO DO PROBLEMA E OBJECTIVOS

Apesar das melhorias significativas das ferramentas BIM em termos de modelação paramétrica na comunicação com a especificação IFC, ainda existem muitas deficiências técnicas sem resolução que impedem uma utilização isenta de erros no que respeita às configurações atribuídas no modelo inicial. Para além dos recursos internos, a qualidade da informação contida num modelo depende também da qualidade do fluxo de dados entre programas e entre disciplinas. A interoperabilidade entre sistemas através do modelo IFC fornece a possibilidade de troca de informações sem que os utilizadores necessitem de conhecimentos aprofundados nesse domínio, contudo, a melhoria da comunicação entre diferentes ferramentas só é possível através do aprofundamento dos conhecimentos nessas matérias. A implementação de dados neutros que aproximam e estimulam a colaboração no projecto, na execução e na operação só se verificará verdadeiramente quando os problemas de interoperabilidade forem suprimidos. Nesse sentido têm sido feitos esforços por parte da BuildingSMART e outras entidades que vão ao encontro destes objectivos, sendo concebidas normas (exemplo da ISO 16739:2013) com vista à sua implementação, de modo a facilitar a interoperabilidade entre *softwares* vocacionados para diferentes actividades durante o ciclo de vida de um edifício.

Na prática corrente de engenharia cada ferramenta pode ter diferentes requisitos para o modo como as representações geométricas são feitas e as informações são indexadas a essas representações ou incorporam o próprio modelo. As capacidades associadas à especificação IFC na compreensão de produtos, relações e recursos da construção, em junção com o seu caracter expansivo tornaram este formato na solução de interoperabilidade de eleição na indústria AEC. Apesar de todas as vantagens trazidas pelo IFC e todos os esforços que vêm sido feitos para que a sua integração seja melhor absorvida pela indústria AEC, ainda existem muitas dificuldades que devem ser ultrapassadas para que se atinjam aceitáveis níveis de satisfação dos utilizadores. (Aram, Eastman, and Sacks 2013). Vários testes e estudos realizados, incluindo o estudo efectuado no Capítulo 3 da presente dissertação, apontam para inúmeros casos onde a informação transferida é deturpada ou até mesmo perdida, diminuindo drasticamente os níveis de confiança no formato IFC

Como principal objectivo este capítulo pretende fornecer indicações e metodologias que possibilitarão a automação de sistemas usando como suporte as classes IFC estudadas.

5.2. ESTRATÉGIA DE ABORDAGEM

Na prática actual as ferramentas de suporte à interoperabilidade são apenas usadas ocasionalmente devido às perdas de informação que se verificam no processo de troca entre *softwares*. Apesar dos elevados custos associados à reintrodução de dados, a impossibilidade de uma correcta transferência de informação para diversas áreas e fases do projecto conduz por vezes a um abandono desses métodos de transferência de dados, agravando o fluxo da comunicação. Estas perdas de informação devem-se em parte ao não comprometimento dos desenvolvedores de programas informáticos para garantir uma correcta transferência de dados e à falta de preocupação para com as questões ligadas à interoperabilidade de sistemas (Bazjanac and Crawley 1998). Na inexistência de um modelo de dados comum, as ferramentas BIM na indústria de construção poderiam apenas comunicar de forma directa entre interfaces que fazem a tradução entre esses formatos, necessitando de múltiplas aplicações que acarretariam custos elevados para o utilizador.

Acreditando nas potencialidades da especificação IFC e no sentido de prestar contributo para um aumento dos níveis de interoperabilidade pelo acréscimo da qualidade de transferência de dados, são sugeridas duas propostas de aplicação que têm como base a implementação das entidades IFC. Uma aplicação pretenderá ser um ponto de partida para a criação semiautomática de modelos de informação e a outra pretenderá ser um meio que possibilita a correcção e manipulação da informação para melhoria da interoperabilidade. Para isso foi necessário um estudo aprofundado no domínio da linguagem e codificação da especificação IFC, investigando as entidades que permitem decompor produtos, estabelecer relações e descrever propriedades e materiais, elementos que poderão constituir um modelo gerado por uma qualquer ferramenta BIM.

O processo base adoptado em ambas baseou-se numa abordagem metodológica, onde primeiro se identificaram individualmente quais os elementos a serem estudados e só depois se procuraram as instâncias IFC que possibilitam a tradução desses elementos para um ficheiro de troca, com um estudo que foi voltado para a definição estrutural dos elementos e outros dados que se detectaram não ser exportados para o modelo IFC. Através desse detalhamento dos elementos individuais será possível fazer o seu agrupamento e indexação de mais informações, depois a multiplicação desses dados permitirá a elaboração de estruturas de edifícios, processo mostrado na Figura 5.1.

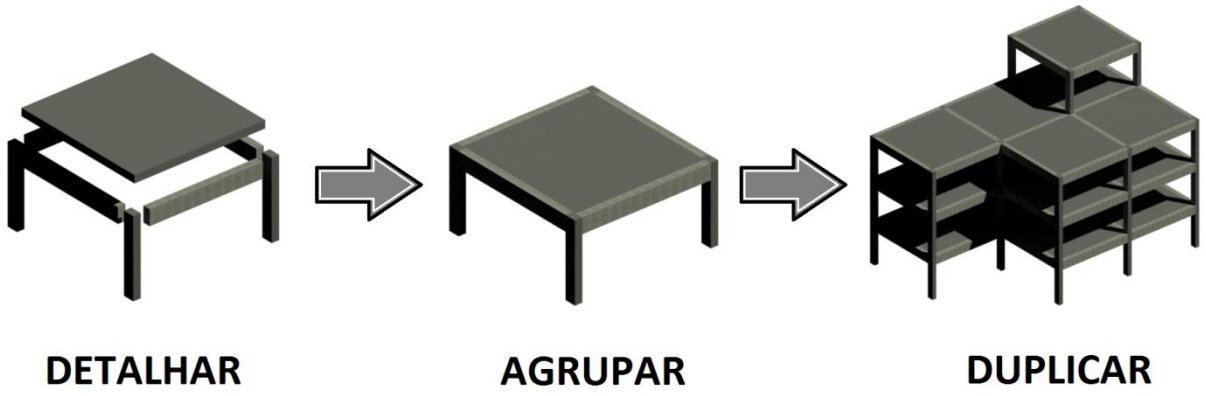


Figura 5.1 – Processo para a definição de elementos compostos.

A informação que atravessa as várias fazes deste processo poderá também ser usada na correcção de algumas lacunas existentes na actualidade e possibilitará a melhoria de processos de troca de informação. Com essas informações poderão ser reproduzidas determinadas sequência de dados,

mantendo a informação mais genérica para uma determinada situação e alterando algumas variáveis, resultando numa multiplicação de definições de objectos, produtos e propriedades. Torna-se por isso necessário ultrapassar a inércia existente em compreender o funcionamento de um ficheiro IFC, pois é só através da sua forma de expressão que é possível usar as entidades IFC para manipular e trabalhar a informação de forma proveitosa para a correcção de erros ou mesmo até a criação de novos dados, tirando-se partido dos conhecimentos apreendidos.

5.2.1. DETALHAR

Esta acção que consiste em detalhar os elementos foi já elaborada e descrita no Capítulo 4. A compreensão de um objecto pela fragmentação dos seus elementos possibilita uma melhor interpretação do verdadeiro sentido e função desse objecto. O conhecimento desses pequenos pedaços de informação permitirão agrupar e duplicar os dados de variadas formas, resultando em inúmeras combinações possíveis para o rearranjo de informações traduzindo os elementos compostos desejados.

Através da Figura 5.2 é possível avaliar a quantidade de fragmentos de informação que são necessários para a definição de um simples pilar através de entidades IFC e ainda assim não nos podemos esquecer que se trata de uma definição parcial, é ainda possível adicionar muitas outras informações que não foram objecto de estudo.

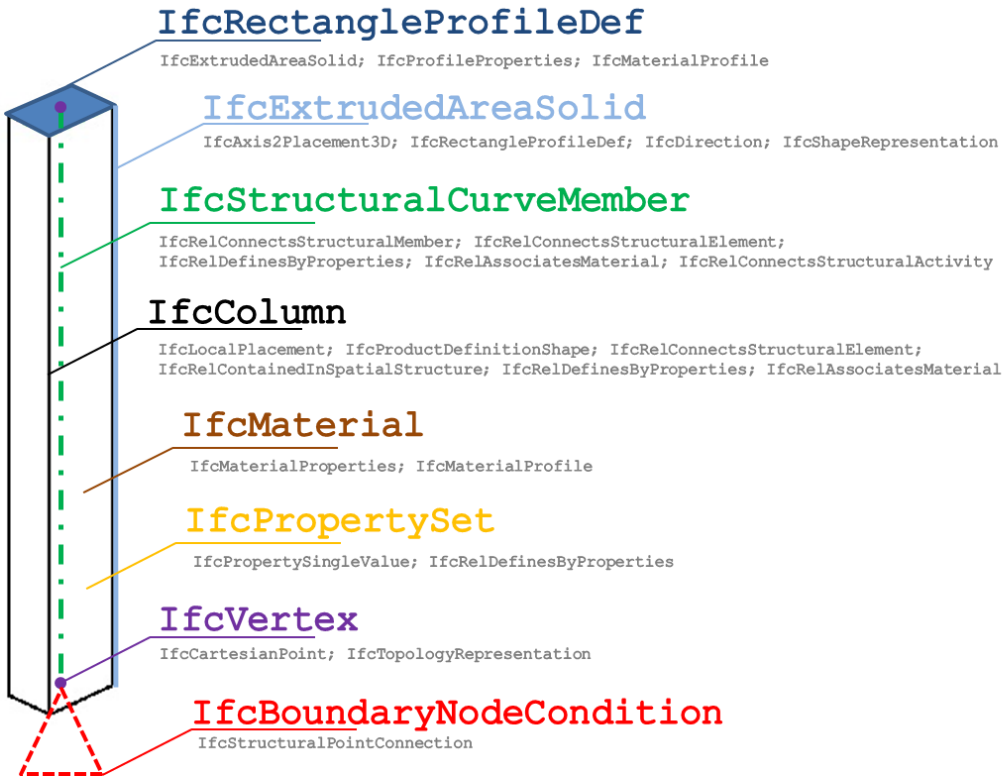


Figura 5.2 – Decomposição de um pilar em classes IFC.

As cores representam as entidades que estabelecem as acções principais para a definição do elemento e as entidades identificadas a cinza identificam algumas entidades que fazem suporte directo à respectiva entidade que define a acção principal. Feita a identificação das entidades necessárias para a definição de um dado objecto, para a elaboração de outros objectos semelhantes mas com outras

configurações apenas será necessário alterar as variáveis fundamentais, como as dimensões, tipo de elemento, materiais e propriedades a atribuir, condições de fronteira, entre outros.

5.2.2. AGRUPAR

O agrupamento das informações é a fase que transforma os fragmentos de informação em dados utilizáveis, munindo-os de significado, isto é, depois de agrupados os diferentes elementos passam a desempenhar um papel específico nesse modelo. Esse agrupamento de classes é feita através da identificação das suas relações. Para além das relações através de atributos a generalização de relação mais abstracta é feita pela entidade *IfcRelationship* que estabelece todas as relações entre objectos dentro do modelo IFC, que poderão ser estabelecidas de um para um ou de um para vários. Desta forma é possível manter propriedades específicas imputadas a um determinado relacionamento, podendo-se mais tarde especificar um comportamento. Os comportamentos são diferenciados pelas diferentes entidades subtipo da *IfcRelationship* e são os seguintes:

- *IfcRelAssigns* – Uma relação de atribuição estabelece uma relação através de um "*link*" pelo meio do qual um objecto pode navegar para outros objectos, através de uma relação bidireccional.
- *IfcRelAssociates* – O relacionamento de associação faz referência às fontes da informação, podendo esta residir internamente ou externamente ao ficheiro de dados, visto não haver dependência implícita nesta associação.
- *IfcRelConnects* – A relação de conectividade faz a conexão dos elementos permitindo estabelecer alguns critérios adicionais, como por exemplo determinadas restrições e condições de fronteira.
- *IfcRelDeclares* – Este tipo de relacionamento permite a declaração da relação entre objectos ou propriedades com um projecto.
- *IfcRelDecomposes* – A composição ou decomposição dos elementos é feita através do conceito geral definido por este tipo de relacionamento. Deste modo obtém-se a habilidade de partir de um todo para as partes e das partes para um todo.
- *IfcRelDefines* – Por último tem-se uma definição mais genérica e abstracta que serve essencialmente para atribuir um tipo de objecto a uma determinada ocorrência, atribuir propriedades a uma instância ou ainda atribuir um modelo de propriedades a um tipo de propriedade.

Para além do conceito de relacionamento, agrupar também inclui o conceito de agregar. Para além da informação necessária para definir um elemento podem-se atribuir ou agregar informações externas a esse elemento, como acontece por exemplo, na atribuição de cargas a um pilar. Essa carga pode ser construída de forma independente ao elemento a que é aplicada, definindo-se algumas das suas variáveis que seriam o ponto de aplicação, direcção, sentido e intensidade e só depois agregar essa carga ao elemento.

Para concluir e voltando ao exemplo do pilar, a Figura 5.3 retirada do site da BuildingSMART demonstra algumas das relações que é possível estabelecer inversamente a um pilar e objectos ou elementos..

```
ENTITY IfcColumn
  INVERSE
    HasAssignments              : SET OF IfcRelAssigns FOR RelatedObjects;
    Nests                       : SET [0:1] OF IfcRelNests FOR RelatedObjects;
    IsNestedBy                  : SET OF IfcRelNests FOR RelatingObject;
    HasContext                  : SET [0:1] OF IfcRelDeclares FOR RelatedDefinitions;
    IsDecomposedBy              : SET OF IfcRelAggregates FOR RelatingObject;
    Decomposes                  : SET [0:1] OF IfcRelAggregates FOR RelatedObjects;
    HasAssociations             : SET OF IfcRelAssociates FOR RelatedObjects;
  INVERSE
    IsDeclaredBy                : SET [0:1] OF IfcRelDefinesByObject FOR RelatedObjects;
    Declares                    : SET OF IfcRelDefinesByObject FOR RelatingObject;
    IsTypedBy                   : SET [0:1] OF IfcRelDefinesByType FOR RelatedObjects;
    IsDefinedBy                 : SET OF IfcRelDefinesByProperties FOR RelatedObjects;
  INVERSE
    FillsVoids                  : SET [0:1] OF IfcRelFillsElement FOR RelatedBuildingElement;
    ConnectedTo                 : SET OF IfcRelConnectsElements FOR RelatingElement;
    IsInterferedByElements      : SET OF IfcRelInterferesElements FOR RelatedElement;
    InterferesElements          : SET OF IfcRelInterferesElements FOR RelatingElement;
    HasProjections              : SET OF IfcRelProjectsElement FOR RelatingElement;
    ReferencedInStructures      : SET OF IfcRelReferencedInSpatialStructure FOR RelatedElements;
    HasOpenings                 : SET OF IfcRelVoidsElement FOR RelatingBuildingElement;
    IsConnectionRealization     : SET OF IfcRelConnectsWithRealizingElements FOR RealizingElements;
    ProvidesBoundaries          : SET OF IfcRelSpaceBoundary FOR RelatedBuildingElement;
    ConnectedFrom               : SET OF IfcRelConnectsElements FOR RelatedElement;
    ContainedInStructure        : SET [0:1] OF IfcRelContainedInSpatialStructure FOR RelatedElements;
```

Figura 5.3 – Relações de atribuição inversa a um pilar. Adaptado de (BuildingSMART 2013).

5.2.3. Duplicar

Uma vez detalhadas as informações fragmentadas e conhecidas as condições para se estabelecerem as relações estão criadas as bibliotecas de dados que permitem a construção de elementos compostos, neste caso estruturas, pela reprodução e combinação desses dados. A presente dissertação inicia uma biblioteca de entidades IFC com as informações disponibilizadas no Capítulo 4. À medida que se vão aprofundando os conhecimentos no domínio do IFC poderão ser adicionadas outras entidades que ainda não constam dessa biblioteca, expandindo dessa forma os ramos de aplicabilidade desses conhecimentos.

Como se vem referindo, uma estrutura é no fundo a composição de múltiplos elementos simples inter-relacionados. A duplicação de um elemento não significa obrigatoriamente a necessidade de duplicação de relações ou atributos. A título de exemplo mostra-se a Figura 5.4 onde duas sapatas partilham uma mesma secção.

```
        #404= IFCRECTANGLEPROFILEDEF (.AREA.,'150x200',#190,1.5,2.0);
        #405= IFCEXTRUDEDAREASOLID (#404,#191,#13,0.75);
           #406= IFCSHAPEREPRESENTATION (#202,'Body','SweptSolid',(#405));
              #407= IFCPRODUCTDEFINITIONSHAPE ($,$,(#406));
              #408= IFCPRESENTATIONLAYERASSIGNMENT ('Footing_Profile_A','No
                                                     description',(#406),$);
     #409= IFCCARTESIANPOINT ((0.0,0.0,-0.75));
        #410= IFCAXIS2PLACEMENT3D (#409,$,$);
           #411= IFCLOCALPLACEMENT (#192,#410);
                 #412= IFCFOOTING ('1we2ARoE50netW0RKs4eva',#5,'Sapata 1',
                                   $,$,#411,#407,$,.PAD_FOOTING.);
     #413= IFCCARTESIANPOINT ((5.5,0.0,-0.75));
        #414= IFCAXIS2PLACEMENT3D (#413,$,$);
           #415= IFCLOCALPLACEMENT (#192,#414);
                 #416= IFCFOOTING ('2we2ARoE50netQ4FGs4drw',#5,'Sapata 2',
                                   $,$,#415,#407,$,.PAD_FOOTING.);
```

Figura 5.4 – Exemplo da partilha de uma secção por duas sapatas.

A possibilidade de partilha de relações e atributos é uma vantagem de grande utilidade pois permite alterar simultaneamente as informações contidas em todos os elementos de um mesmo tipo pela alteração das variáveis numa única instância e permite também uma economia e eficiência na utilização da linguagem de código.

5.3. PROPOSTAS DE IMPLEMENTAÇÃO

5.3.1. EXPANSÃO DA APLICAÇÃO "*EVOLUTION BIM PAVILION*"

Uma das componentes que incentivou o estudo da linguagem IFC foi a procura de entidades que viabilizassem a expansão de uma aplicação informática que tem como suporte base a codificação IFC para o desenvolvimento de estruturas. A aplicação em causa é denominada "*Evolution BIM Pavilion*" e foi inicialmente desenvolvida pelo Prof. Dr. Miguel Ângelo Carvalho Ferraz. Esta ferramenta permite a modelação automática de estruturas apenas pela introdução das variáveis em caixas de introdução de dados (*Input Box*), Algumas dos separadores de informação que fazem parte da interface do programa são apresentadas na Figura 5.5.

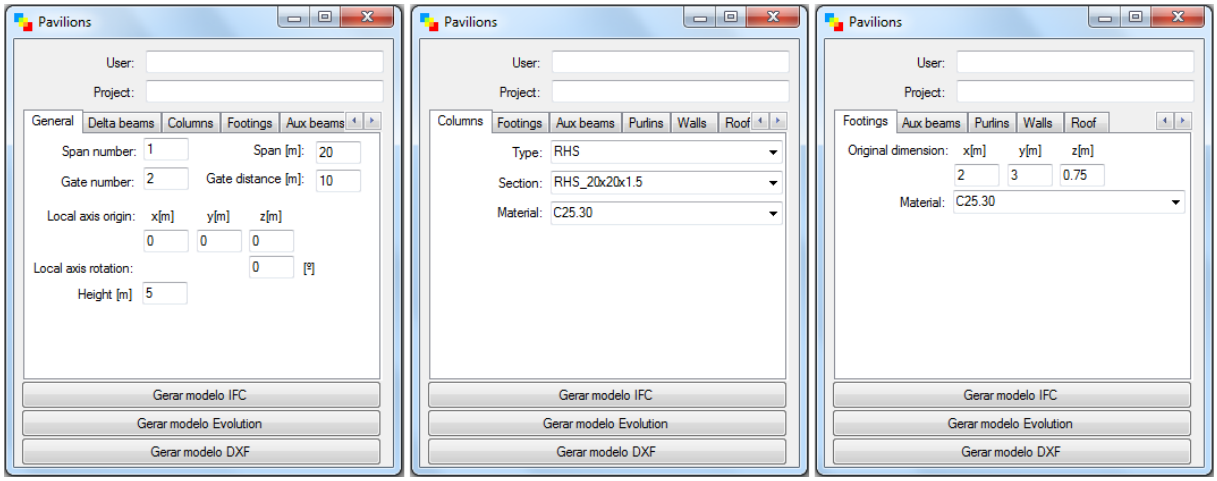


Figura 5.5 – Aplicação *Evolution BIM Pavilion*.

Depois de introduzida a informação respectiva aos elementos que irão compor a estrutura pretendida a aplicação gera um modelo IFC que depois poderá ser aberto por uma qualquer ferramenta BIM que já integre o modelo IFC. A versão inicial desse *software* permitia a elaboração de estruturas de pavilhões de um ponto de vista geométrico com a indicação dos materiais mas apenas através de etiquetas, sem nenhuma materialização de propriedades físicas ou outras características. Os pavilhões gerados pela aplicação na versão disponível no momento são constituídos essencialmente por vigas delta, vigas auxiliares e pilares, existindo a possibilidade de incluir sapatas, paredes e telhados. A Figura 5.6 demonstra a visualização do pavilhão gerado com os dados introduzidos na Figura 5.5. Pretende-se agora vocacionar esta aplicação para a elaboração de estruturas porticadas, construídas pela composição de sapatas, pilares vigas e lajes. Para além disso pretende-se também possibilitar a atribuição de cargas pontuais, lineares e uniformemente distribuídas a um determinado elemento da estrutura, bem como fazer uma atribuição dos materiais definindo propriedades físicas.

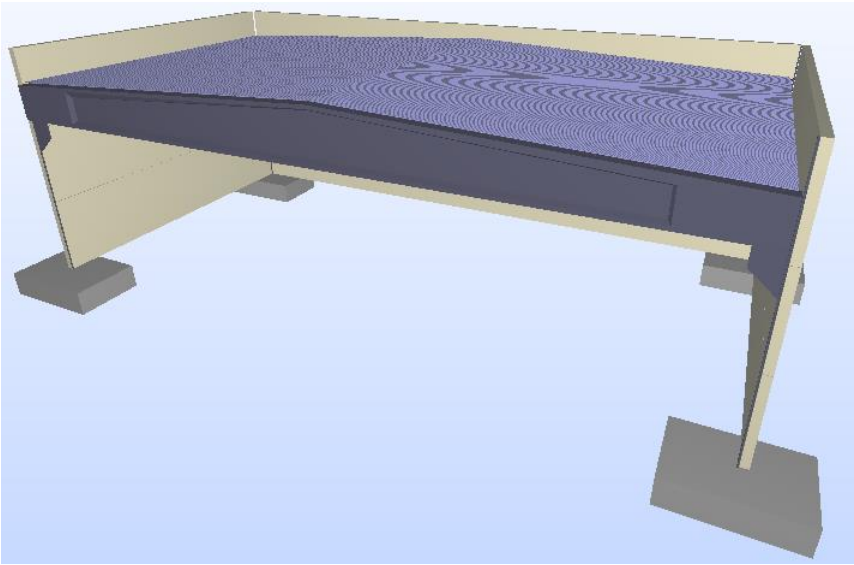


Figura 5.6 – Pavilhão originado pela aplicação *Evolution BIM Pavilion.*

Utilizando como guia alguma da informação originada pela aplicação na criação de modelos, foi possível redefinir algumas entidades geométricas para a elaboração de estruturas porticadas, às quais foram adicionadas informações analíticas e mecânicas para reforçar e melhorar a definição dos elementos usados. Este estudo que foi já concretizado no Capítulo 4 constitui a proposta para a implementação de novas entidades para dotar o programa de novas capacidades. A Figura 5.7 apresenta a interface da nova aplicação que resultou da implementação de entidades referida, que agora possibilita a elaboração de estruturas de edifícios regulares.

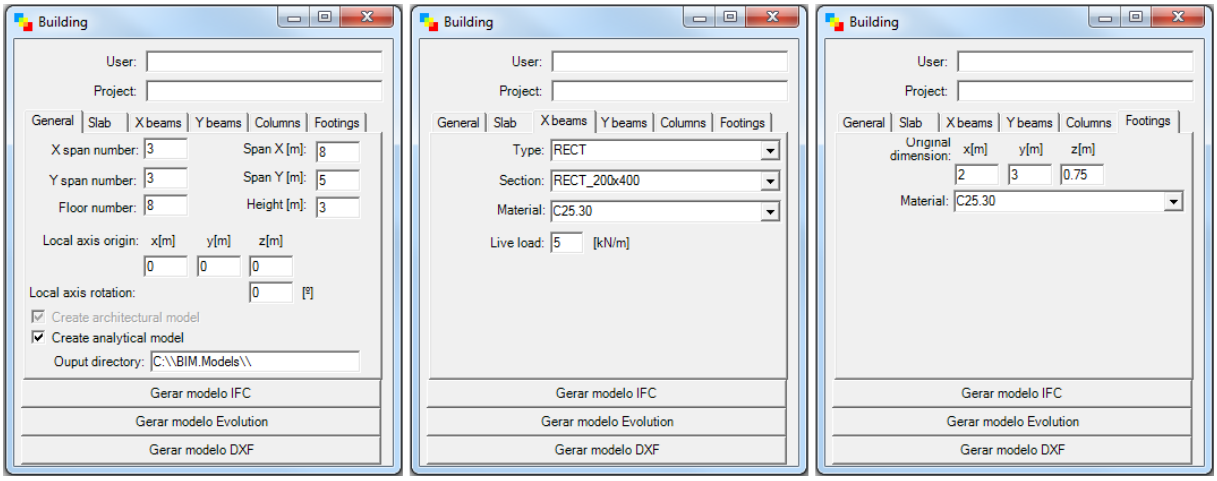


Figura 5.7 – Aplicação *Evolution BIM Buildings*.

Para além de produtos e propriedades também é indispensável a definição de relações. A junção das diferentes vertentes geométricas, analítica e mecânica num só modelo é feita pela atribuição de relações e por isso podem ser gerados modelos compostos apenas por um tipo de informação ou pela combinação e junção de vários tipos de informação. A possibilidade de seleccionar a informação que incorpora o modelo IFC gerado, permite optar pela criação de um modelo com um âmbito mais alargado ou criação de um modelo apenas com a informação estritamente necessária. O primeiro tem a vantagem de fazer uma definição mais completa dos elementos e o segundo tem como vantagem evitar

que o ficheiro criado contenha um excesso de informações que podem não interessar ao destinatário e até dificultar a sua percepção, pois normalmente este tem uma vocação diferente do emissor. Portanto, como mostrado na Figura 5.7, a nova aplicação remodelada possibilita pela validação da *check box* "Create analytical model" a criação ou não do modelo analítico da estrutura. A definição analítica permite estabelecer o relacionamento entre os elementos que compõem o edifício e auxilia o programa destino a compreender o funcionamento dessa estrutura.

A Figura 5.8 apresenta a estrutura originada com os dados introduzidos na Figura 5.7, mostrando à esquerda o modelo com a sua definição arquitectónica (visualizado com o programa Solibri Model Viewer) e à direita o modelo analítico onde se aplicou uma carga linear de 5 kN/m às vigas na direcção X (visualizado com o programa Constructivity Model Viewer).

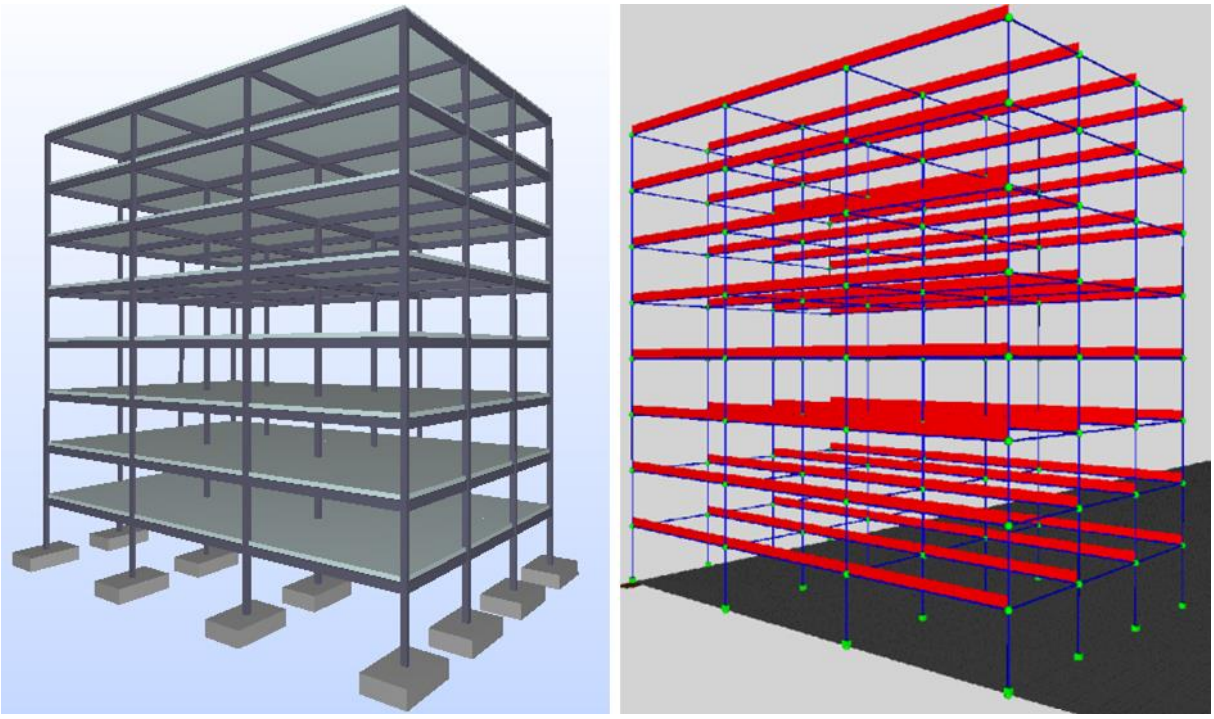


Figura 5.8 – Estrutura originada pela aplicação *Evolution BIM Buildings*.

5.3.2. Desenvolvimento de Aplicação para a Restituição de Dados ao Ficheiro IFC

Os relatos apresentados no Capítulo 3 deram conta de que na realização de troca de dados usando o modelo IFC são perdidas informações valiosas. Como também já foi constatado nos casos expostos nesse capítulo, uma dada aplicação destino ao importar um ficheiro IFC faz a verificação das informações que são recebidas como etiquetas (*labels)*. Ao serem enviados identificadores através de etiquetas do tipo *IfcIdentifier*, *IfcText* ou *IfcLabel* contido nas entidades IFC é possível recuperar informação perdida, pois sabe-se qual a informação que foi perdida e qual a informação original correspondente que deveria ter sido enviada pelo programa origem. Através desses fragmentos de informação torna-se possível fazer a re-caracterização dos dados iniciais ou até mesmo fazer um acréscimo de informações complementares. Para a identificação destas etiquetas e de outras entidades IFC relevantes que poderão ser indicadores de pontos onde é necessário o reforço de informação poderá ser usado o programa *IFC File Analyser* do NIST que faz a leitura de ficheiros IFC e constrói um ficheiro Excel com as entidades presentes nesse ficheiro IFC, reconhecendo as principais relações e atributos dessas instâncias

Tomemos o exemplo dos materiais para melhor se compreender o processo de identificação destas *labels*. Se numa entidade *IfcMaterial* estiver contida a informação "Concrete C20/25" que é identificada internamente como informação do tipo *IfcLabel*, o programa destino entende como sendo uma solicitação para atribuir um material com essa designação e procura na sua base de dados um material que corresponda ao solicitado para fazer a requerida atribuição. A partir daí resultam duas situações, ou é encontrado o material correspondente e feita a respectiva atribuição, ou não é encontrado o material correspondente e criado um novo material sem propriedades (no caso de não estarem quaisquer *Property Sets* associadas ao material). A primeira situação é a situação desejável mas poucas vezes verificada porque normalmente a forma como os nomes dos materiais são escritos ou apresentados variam de programa para programa. Na segunda situação onde não há correspondência entre o nome contido numa etiqueta do ficheiro IFC e o nome de um material da base de dados do programa destino sendo gerado um novo material é uma situação recorrente. Por forma a contornar este e outros problemas idênticos propõe-se a metodologia de resolução apresentada na Figura 5.9.

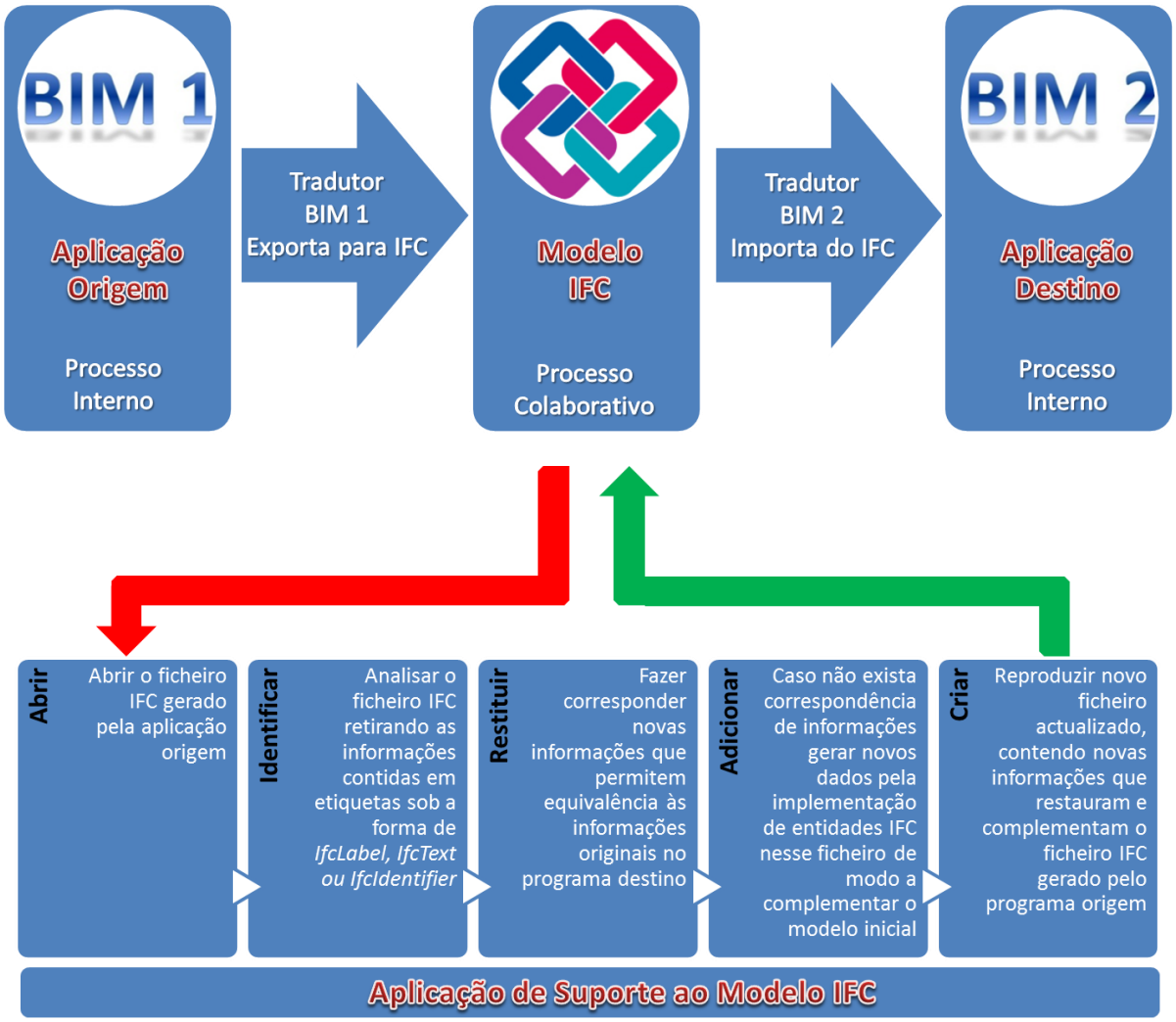


Figura 5.9 – Processo de transferência de informação entre duas aplicações usando o modelo IFC e uma aplicação de suporte.

Esta solução que se apresenta passa pela realização de uma aplicação que opere em paralelo com o modelo IFC logo depois deste ser criado. Como não é possível modificar ou fazer qualquer tipo de alterações na tradução que é feita nas aplicações origem e destino, a única fase onde é possível intervir é nessa fase compreendida entre a exportação e a importação quando se obtém o modelo IFC gerado pela aplicação BIM 1.

A Figura 5.10 expõe um fluxograma que traduz o funcionamento da aplicação. No caso da informação acerca dos materiais (mas também outros tipos de informações) a primeira condição imposta é garantir que o nome da etiqueta do material se mantém intacto. Depois o programa deve identificar o programa origem (*input*) e o programa destino (*output*), bem como todos os materiais contidos nesses programas. Com uma base de dados robusta das propriedades usadas nesses programas que foram utilizados na troca de ficheiros deve-se fazer corresponder os materiais que são equivalentes nos diferentes programas, por exemplo, o Material X na aplicação BIM 1 corresponde ao Material Y na aplicação BIM 2. Ao executar estas informações o que o programa irá fazer é simplesmente trocar o nome contido na etiqueta do *IfcMaterial* por um nome que o programa destino conheça. No caso de não existir um material no programa BIM 2 que corresponda ao material exportado pelo programa BIM 1, a aplicação de suporte poderá criar um novo Material Z que faça a correspondência ao Material X, pela adição das propriedades desse material através de instâncias IFC e *PropertySets*.

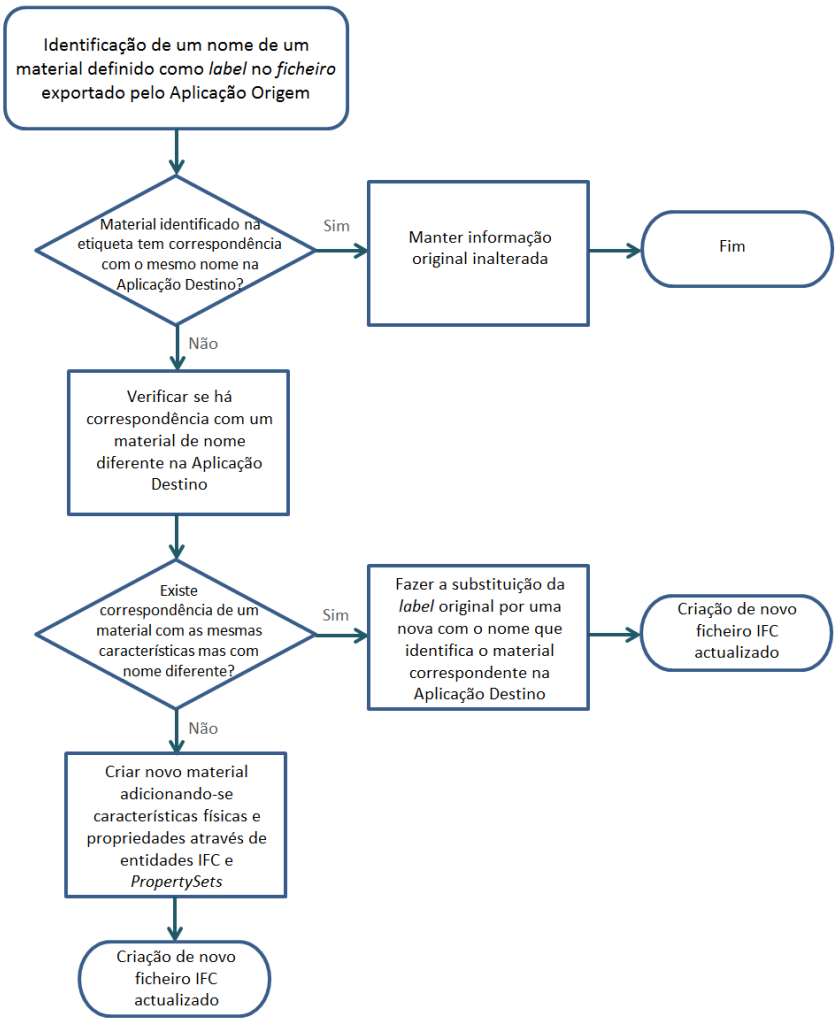


Figura 5.10 – Fluxograma de acréscimo de valor ao ficheiro IFC exportado pela adição de informações.

Da mesma forma que se procedeu a restituição dos materiais poder-se-á fazer a restituição de outras informações. A definição dos componentes estruturais através da sua definição analítica é outro dos objectivos desta aplicação. As vantagens associadas à atribuição de representações de modelos analíticos está relacionada com a definição desses objectos por elementos infinitamente rígidos, por forma a garantir que elementos que concordam num mesmo nó permanecem conectados após a importação do ficheiro IFC. Enquanto que na situação dos materiais é necessário identificar uma etiqueta para fazer a atribuição do material pretendido, no caso das definições analíticas dos elementos, estas poderão ser inseridas pela aplicação de suporte independentemente de terem sido ou não um dos dados constantes no ficheiro original, isto porque a definição analítica da estrutura será feita pelo reconhecimento das secções dos elementos, informação que geralmente consta no ficheiro IFC exportado, mesmo quando os níveis de interoperabilidade são considerados baixos. Apesar desta definição analítica ter uma implementação mais trabalhosa, percebem-se as vantagens que daí advêm, já que contribui activamente para um aumento dos níveis de interoperabilidade com programas vocacionados para o cálculo estrutural.

Desde que se consigam enviar para o IFC informações elementares através de entidades genéricas ou até mesmo *Labels*, a aplicação será capaz de reconstruir a informação original, vencendo algumas das dificuldades de interoperabilidade entre programas. O factor chave para o sucesso desta aplicação reside numa base de dados de entidades a mais completa quanto possível e uma boa base de dados dos elementos usados pelos programas e um entendimento da forma como eles operam.

5.4. DESENVOLVIMENTOS FUTUROS

O estudo semântico e sintáctico do IFC exposto no capítulo anterior permitiu a extracção de informação valiosa para a definição de modelos estruturais e aplicação das entidades IFC estudadas. As entidades expostas apenas correspondem às entidades cujo estudo foi feito de forma completa, outras entidades surgiram mas o tempo de investigação limitado não permitiu um estudo completo e por isso não foi possível a sua aplicação ou demonstração de exemplos. O ficheiro IFC elaborado (em anexo) procurou abranger as áreas da arquitectura, estruturas e mecânica, mas devido à abrangência da especificação IFC continua a ser necessário um grande aprofundamento destas matérias. É recomendado que na elaboração de modelos a partir da linguagem IFC se tenham sempre presentes as diferenças entre estes três ramos do BIM e também as suas relações (Figura 5.11).

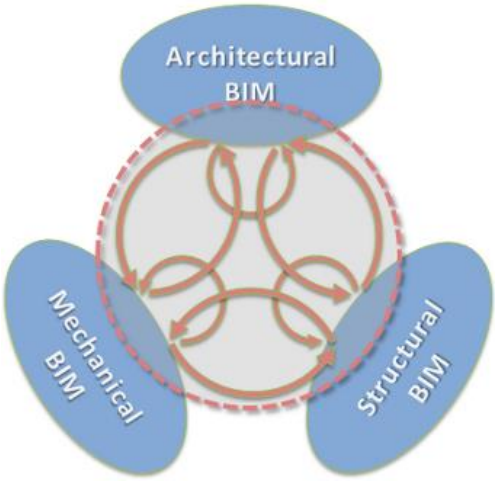


Figura 5.11 – Relação entre três vertentes BIM (BuildingSMART 2013)

Sugerem-se então uma lista com algumas entidades que poderão permitir a continuidade e aprofundamento deste estudo. São entidades que poderão permitir a inclusão de outros tipos de componentes da construção, aberturas em elementos (por exemplo a abertura numa laje para colocação das escadas ou abertura numa parede para colocação de portas e janelas), definição de outros tipos de secções, definição de sistemas construtivos, ligação de elementos com excentricidade, entre outras.

- *IfcWall; IfcStair; IfcDoor; IfcWindow*
- *IfcRelVoidsElements; IfcOpeningElement*
- *IfcParameterizedProfileDef;*
- *IfcStyledItem*
- *IfcElement; IfcBuildingElement*
- *IfcMaterialLayer; IfcMaterialLayerSet; IfcMaterialLayerSetUsage*
- *IfcVertexLoop*
- *IfcStructuralLoadTemperature*
- *IfcBuildingSystem;*
- *IfcRelServicesBuildings*
- *IfcRelConnects; IfcRelConnectsWithRealizingElements*
- *IfcConnectionPointEccentricity; IfcRelConnectsWithEccentricity*

6
CONCLUSÕES

6.1. DISCUSSÃO

Uma das principais conclusões que este trabalho permite retirar é que para aumentar os níveis de eficiência da indústria AEC é necessário aumentar os níveis de interoperabilidade. Foram mostrados problemas gerados por uma interoperabilidade deficiente que cada vez mais se revela insuficiente face às exigências actuais, foram estudadas entidades IFC para aprofundar os conhecimentos neste domínio que permitem perceber o seu modo de funcionamento e foi exposta uma sugestão de aplicação da especificação IFC que se pretende que tenha aplicabilidade num futuro próximo. Espera-se que com a implementação da norma ISO 16739:2013, que fomenta e potencializa o IFC enquanto formato aberto para a interoperabilidade entre ferramentas BIM, se venha a verificar uma melhoria quer na quantidade de informação que é possível transferir entre programas, quer na qualidade dos moldes em que essa transferência é feita.

Apesar dos esforços feitos pela BuildingSMART, se os distribuidores de *softwares* continuarem a desperdiçar a maioria das potencialidades do IFC e usarem apenas este formato para a troca de informações básicas e de baixa complexidade, significa que não há uma aceitação do formato IFC por parte destes, o que poderá conduzir o IFC para uma queda no domínio do obsoleto. Se essa realidade se verificar então significa que dificilmente se chegará a um entendimento onde se concretize a aceitação de um formato de troca de dados aberto, o que representaria uma enorme perda de todo o potencial e vantagens que esse formato traria e já trouxe para a indústria AEC. Nesta situação hipotética, existiriam poucas soluções para os problemas de interoperabilidade que se sentem na actualidade e a resolução desses problemas seria só exequível se não fosse necessário realizar trocas de informação entre diferentes sistemas informáticos. Isto apenas será concretizável quando existir um avanço suficientemente grande das TIC e das ferramentas BIM para que seja possível criar um sistema informático que inclua todos os âmbitos dos BIM num único *software*, deste modo todos os intervenientes num projecto e no ciclo de vida de um edifício utilizariam a mesma ferramenta e as questões de troca de informação estariam ultrapassadas pois essa troca já não seria necessária, contudo e ainda assim, isto só seria verdade se todos os envolvidos usassem o mesmo *software* de um dado distribuidor. Conclui-se portanto que se um documento não for passível de troca então este tem valor limitado (Jernigan 2008).

A noção de *Building Information Modeling* assenta neste conceito de universalidade. Mas o BIM não está completo, para que se torne verdadeiramente universal não basta que permita a inclusão de varias vertentes do projecto num único modelo virtual, é necessário que o seu âmbito seja ainda mais alargado abrangendo recursos, processos e fundamentalmente pessoas durante todo o ciclo de vida da construção admitindo um fluxo de informação continuo. Devem então ser tomadas acções que vão ao

encontro de uma melhor compreensão dos objectivos dos BIM e que resultem numa melhor integração prática dessas ferramentas. Para isso é necessário promover a aliança entre empresas desenvolvedoras de *softwares* que têm no fundo os mesmos objectivos, facilitando a exploração de novos caminhos de evolução tecnológica, usando-se a inovação para a gestão de informação. É também necessário educar e informar, apostando-se na divulgação dos BIM para que os seus objectivos sejam percebidos por toda a comunidade da indústria AEC e não apenas por uma pequena parte dessa indústria. Desta forma será possível aplicar as competências certas, no momento certo pelo elemento indicado, o que por si só resultará num grande aumento da eficiência e produtividade.

6.2. PRINCIPAIS DIFICULDADES

A elaboração da presente dissertação caracterizou-se pelas grandes dificuldades iniciais em compreender o funcionamento linguístico da especificação IFC. Por essa razão o estudo realizado foi fortemente encaminhado com o propósito de minimizar o impacto provocado pelo primeiro contacto com estas matérias, constituindo um guia introdutório que pretende também ser resposta para a escassez de demonstração prática e introdução de conceitos IFC. O próprio conceito BIM começa só agora a ser mais divulgado e por isso ainda são poucos os que possuem conhecimentos de programação em código IFC. Neste estudo valeram todas as preciosas indicações dadas pelos orientadores, como a indicação de onde procurar informação e sobretudo como procurar, possibilitando um aprofundamento do conhecimento das matérias tratadas.

Uma das primeiras dificuldades foi a dificuldade de acesso às principais normas. Os elevados preços associados tornaram a sua aquisição formalmente impraticável. Ainda assim foi possível encontrar conteúdo parcial das principais normas usadas ou até mesmo normas completas mas noutras línguas que não o português ou o inglês, sendo necessário um trabalho acrescido na tradução desses documentos.

Outra limitação com a qual se conviveu foi a restrição de acesso aos *softwares*. Muitos desses problemas foram resolvidos pela requisição de licenças de estudante, no entanto o acesso a determinados softwares que poderiam potencializar esta investigação não são acessíveis pela comunidade estudante ou são apenas parcialmente acessíveis em versões temporárias com inúmeras funcionalidades bloqueadas, como é exemplo o programa Constructivity.

Os problemas encontrados na exportação e importação de modelos IFC nos programas testados também foram uma barreira ao próprio estudo da linguagem IFC. A triagem das informações enviadas para um ficheiro IFC e a desorganização dessa informação gerada dificultaram a aprendizagem do código IFC, pois para além da escassez da demonstração de casos de aplicação, os próprios casos práticos não constituíam exemplos completamente fidedignos. Se os programas fizessem uma exportação que traduzisse o modelo virtual por completo sob a forma de linguagem IFC, esses modelos gerados constituiriam bons exemplos de aplicação e permitiriam retirar o "*know-how*" pela observação, compreensão e análise do encadeamento de entidades.

No que toca a linguagens de programação é necessária uma atenção redobrada para os pormenores. A troca de um ponto por uma vírgula, a confusão entre uma letra "I" e um número "1", a repetição da numeração do identificador único (#n) em entidades provocando conflitos na leitura, a colocação de atributos num campo errado, a atribuição inversa de uma entidade que não a pressuposta, a criação de um GUID com mais de 22 caracteres impossibilitando o funcionamento da instância, são apenas alguns dos erros frequentes que podem originar grandes contratempos.

6.3. Considerações Finais

A existência de ferramentas e programas de *software* BIM, é ainda pouco explorada. A partir da Figura 6.1 podemos identificar os vários níveis de evolução das ferramentas de auxilio ao projecto, desde os desenhos em formato papel até um previsão daquilo que serão as ferramentas BIM no futuro. Apesar do diagrama de maturidade BIM não conseguir identificar com exactidão que rumo irá o BIM tomar no futuro, esse também não é um aspecto importante. O importante é seguir o caminho da implementação, com uma mente aberta à inovação e às possiveis mudanças que podem surgir na evolução dos processos. Os BIM são já uma realidade que aos poucos vai marcando presença na industria AEC. No nível 0 de maturidade encontram-se as ferramentas CAD que usam processos informatizados de desenho simples como linhas, formas e texto cujo conceito não difere dos tradicionais desenhos em formato papel. No nível 1 de maturidade verificam-se já manipulação de informação e interpretação de processos que auxilia na execução de desenhos bidimensionais mas também tridimensionais. Na maturidade nível 2 encontram-se os conceitos associados ao BIM, sistema que possibilita a realização de desenhos mas que não se representam através de linhas e formas básicas, mas sim através de dados que possibilitam descrever objectos virtualmente pela definição das suas caracteristicas e propriedades físicas que podem ser operadas e interpretadas por pessoas ou até mesmo por sistemas robotizados. A interoperabilidade apresenta-se como o factor principal que possibilita a passagem para o nível 3 de maturidade, onde a comunicação entre profissionais e integração de modelos ao longo do ciclo de vida da construção será uma constante. (Sinclair 2012).

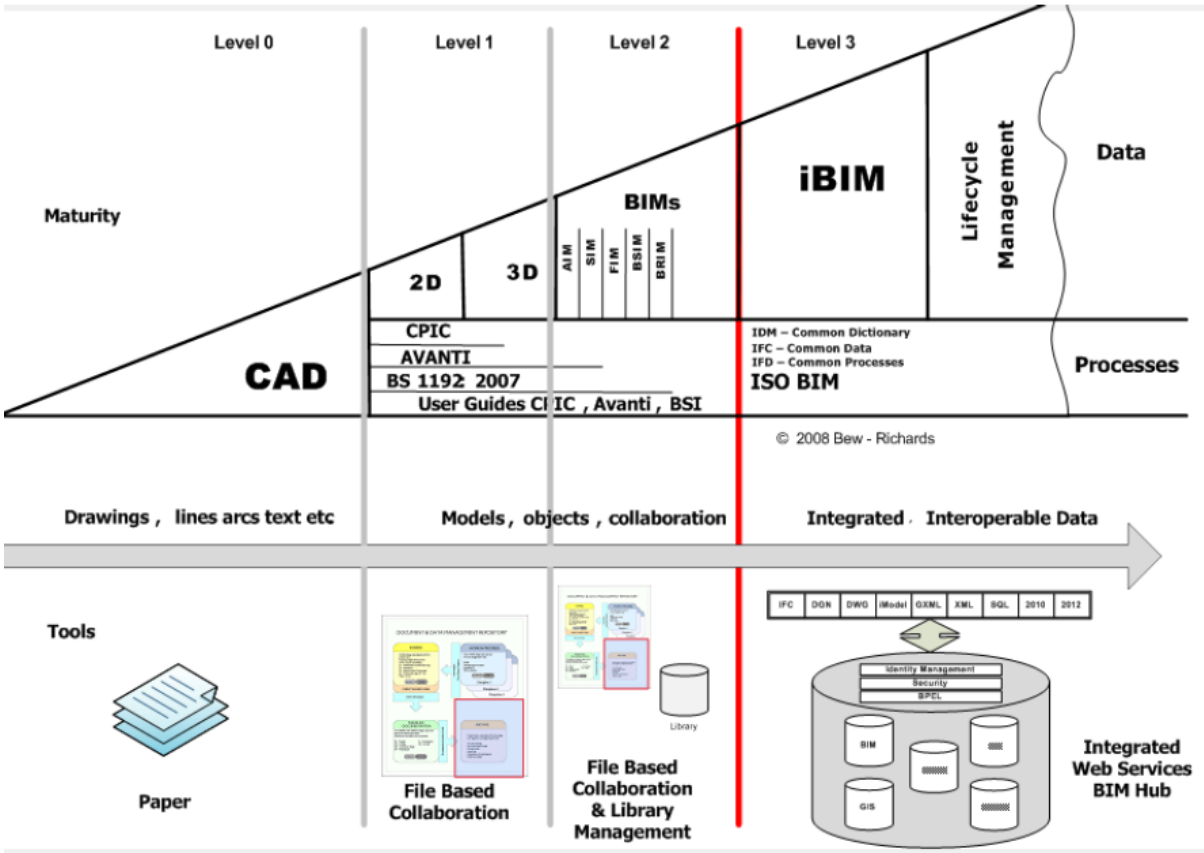


Figura 6.1 – Diagrama de maturidade BIM (Bew and Richards 2008).

No presente, os profissionais desta indústria ainda não tiram o máximo partido destas ferramentas e por isso o seu potencial ainda não é maximizado, ainda são poucos os que beneficiam das dimensões 4D (controlo de prazos) e 5D (controlo de custos) oferecidas por estas ferramentas. Conclui-se portanto que na prática quotidiana nos encontramos situados à esquerda da linha vermelha. É necessário formar os profissionais da indústria da construção para que saibam usufruir das capacidades dos BIM e as apliquem à prática construtiva através de métodos inovadores (BIM Management for value 2011).

A plena implementação do conceito BIM trará os seguintes benefícios:

- Reduz erros e omissões nos documentos de construção;
- Reduz situações de conflito;
- Reduz a necessidade de refazer trabalhos;
- Reduz o tempo de execução de projecto e de tarefas;
- Permite melhor compreender a construção;
- Permite a inovação de processos;
- Motiva o trabalho colaborativo;
- Menos desperdício de recursos;
- Oferece novos serviços e novas vantagens para os clientes;
- Aumenta a qualidade geral do projecto.

"BIM is not just a 3D visual model. It's possible to have a model without any geometric information but with lots of other data, and it will still be a Building Information Model."

- *Arto Kiviniemi, Professor of Digital Architecture Design School of the Built Environment, University of Salford*

Uma das conclusões fundamentais é a importância crescente e necessidade em garantir a interoperabilidade de sistemas, pois se estes não possuírem uma interoperabilidade com uma base sólida, a actividade de colaboração e cooperação tornar-se-á ainda mais dificultada ou até mesmo impraticável.

Estes são alguns dos benefícios da interoperabilidade numa situação ideal:

- Elimina a necessidade de reintrodução de dados já construídos.
- Elimina a necessidade de troca de informações em formato papel.
- Permite que todos os membros de uma equipa trabalhem sobre um mesmo ficheiro e desse modo evita-se a duplicação de ficheiros e melhora-se a precisão do modelo já que todos os elementos da equipa trabalham sobre a mesma base.
- Aumenta a rapidez da elaboração do projecto.
- Reduz a propagação de erros e falhas.
- Permite a reutilização da informação ao longo do ciclo de vida do projecto.

- Alarga a visão que as empresas têm dos mercados.
- Diminui os custos de comunicação.

O uso das ferramentas BIM apenas para modelação tridimensional tem apesar de tudo a vantagem de indirectamente começar a promover aos seus utilizadores o uso e a importância das outras competências desses utensílios. Mesmo os indivíduos que não estavam a par das potencialidades que esses mesmos instrumentos poderiam ter, estão agora mais sensíveis para a verdadeira filosofia BIM e assim sendo começa-se já a verificar uma maior abertura de mentalidades e um despertar crescente de interesses por estas matérias. A generalidade dos profissionais começa agora a olhar para as ferramentas BIM como inevitáveis sucessoras das actuais ferramentas CAD e outras ferramentas complementares. É certo que BIM fará parte do futuro, apenas continua a existir a dúvida de como será utilizado e implementado, sabe-se apenas que deve integrar as necessidades dos utilizadores e todo o processo construtivo.

As soluções tecnológicas poderão passar pela implementação dos formatos abertos normalizados que devem ser adoptados universalmente por toda a indústria de *softwares* da construção. Admite-se que assim as empresas distribuidoras de *software* perderiam vantagens comerciais no que toca à salvaguarda de informações de caracter privado, mas desse modo toda a indústria AEC ficaria a ganhar não só em qualidade de troca e partilha de informação mas também em inovação técnica e metodológica. Como vimos, esforços estão a ser feitos nesse sentido, inclusive através da normalização ISO, para que através da especificação IFC se consiga estar um passo mais perto dessa realidade, contudo o caminho da interoperabilidade poderá não depender apenas de uma norma mas de várias. O certo é que depois de uma década e meia de implementação o formato IFC continua a não ser tão eficaz como os utilizadores esperavam que fosse e por isso ainda não atingiu um nível satisfatório. Os utilizadores continuam a deparar-se com novos pontos onde são necessárias correcções para o uso prático. Conclui-se portanto que os responsáveis pelo desenvolvimento de *softwares* devem primeiro focar-se no melhoramento da qualidade das informações que já são transferidas e só depois alargar as extensões do modelo passiveis de exportação.

Enumeram-se algumas barreiras que impedem uma melhor implementação e progresso da implementação da especificação IFC:

- Os *softwares* fornecem poucas funcionalidades para a troca de informação usando ficheiros de troca IFC. A maioria dos programas que admitem a especificação IFC apenas fornecem uma compatibilidade parcial com este formato de troca, não oferecendo muitas possibilidades de controlo da informação que é transferida.

- Os *softwares* não fornecem especificamente o contexto de decomposição em que se inserem os diferentes tipos de relações de produtos e objectos. Seria vantajoso se um dado programa identifica-se todas as entidades IFC que foram necessárias para traduzir um dado elemento.

- Os *softwares* contêm ainda bastantes erros e problemas na exportação e importação de ficheiros IFC. Esta situação só poderá ser resolvida se houver um esforço e vontade conjunta dos desenvolvedores de sistemas informáticos, associando-se à BuildingSMART para ultrapassarem estas barreiras

- O IFC não permite identificar que representações ou alterações que foram feitas entre dois modelos, o que torna necessário o uso de um programa que permita analisar o ficheiro IFC permitindo comparar as modificações mais significativas (como por exemplo o IFC File analyser do NIST).

- Apesar de ter já um âmbito bastante abrangente, ainda existem elementos e funcionalidades que não são suportadas pelo formato IFC, consequência do amplo domínio da construção.
- A indústria da construção ainda não está suficientemente preparada.

Na situação actual, no que toca a ferramentas BIM, os líderes de mercado moldam a indústria e por isso a única forma de mudar essa filosofia passa pela consideração dos níveis de interoperabilidade por parte dos membros da equipa de projecto aquando da escolha de um dado *software*. Esse deveria ser um dos primeiros factores de decisão de escolha dada a importância que a interoperabilidade terá no decorrer de todo o projecto e as vantagens que trará, pois mesmo que os benefícios não sejam sentidos logo numa fase inicial, serão sentidos em fases posteriores.

Um aumento dos níveis de competição no mercado tecnológico também poderia fomentar a necessidade de tornar os *softwares* mais interoperáveis. Uma maior competição produz mais pressão nas empresas distribuidoras de *softwares* o que dessa forma poderá resultar numa evolução do mercado.

"The biggest innovations in construction industry are more driven by clients than industry itself".

- Paul Morrell, former UK Government's first Chief Construction Adviser

REFERÊNCIAS

REFERÊNCIAS BIBLIOGRÁFICAS

AIA. 2009. Interoperability Position Statement. http://www.aia.org/aiaucmp/groups/aia/documents/pdf/aiab082297.pdf.

Aram, Shiva, Charles Eastman, and Rafael Sacks. 2013. Automation in Construction. *Requirements for BIM platforms in the concrete reinforcement supply chain*: 17, www.elsevier.com/locate/autcon.

Autodesk. *Autodesk* 2013 [cited 22-04-2013. Available from http://www.autodesk.com/.

Bazjanac, Vladimir, and Drury B. Crawley. 1998. "The implementation of Industry Foundation Classes in simulation tools for the building industry."

Bentley Systems, Incorporated 2008. IFC Position Paper 5, http://ftp2.bentley.com/dist/collateral/whitepaper/Building_IFC_Position_Paper_whitepaper_eng.pdf.

Bernstein, Harvey M., Norbert W. Young Jr., and Stephen A. Jones. 2007. "Interoperability in Construction Industry." *McGraw Hill Construction*:36.

Bew, Mark, and Mervyn Richards. 2008. "Building Information Modelling Maturity Levels."

BIM Management for value, cost & carbon improvement. 2011. A report for the Government Construction Client Group. In *Building Information Modelling (BIM) Working Party Strategy Paper*.

BuildingSMART. *International home of openBIM* 2013. Available from http://www.buildingsmart.org/.

Dimyadi, Johannes A. W., Michael Spearpoint, and Robert Amor. 2007. Generating fire dynamics simulator geometrical input using IFC-based Building Information Model. 16.

Eastman, Charles M. 1999. *Building Product Models: Computer Environments Supporting Design and Construction*: CRC Press, Inc.

Eastman, Chuck, Paul Teicholz, Rafael Sacks, and Kathleen Liston. 2011. *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors*. Second Edition ed: Wiley Publishing.

Firmenich, Berthold, Mohamed Nour;, Torsten Richter;, and Christian Koch. 2006. A versioned IFC database for multidisciplinary synchronous cooperation. In *Joint International Conference on Computing and Decision Making in Civil and Building Engineering*. Montréal, Canada.

Gequaltec. 2013. *Wiki da Construção*. FEUP 2011 [cited 13-03-2013 2013]. Available from http://paginas.fe.up.pt/~gequaltec/w/index.php?title=P%C3%A1gina_principal.

Graphisoft. *Graphisoft* 2013 [cited 01-06-2013. Available from http://www.graphisoft.com/images/documentation/documentation-virtual_building.jpg.

Gunter, C.A. 1992. *Semantics of programming languages: structures and techniques*: London.

Hipólito de Sousa et al., João Poças Martins, André Monteiro. 2011. Projecto SIGABIM. In *GEQUALTEC*, edited by Departamento de Engenharia Civil Secção de Construções Civis, Faculdade de Engenharia da Universidade do Porto (FEUP).

IAI. 1999. Industry Foundation Classes - Realese 2.0. In *IFC Object Model Architecture Guide*, edited by Thomas Liebich.

ISO. 2002. ISO 10303-21:2002. In *Part 21: Implementation methods: Clear text encodind of the exchange structure*.

ISO. *International Organization for Standardization* 2013. Available from http://www.iso.org/.

Jernigan, F.E. 2008. *Big BIM, Little Bim: The Practical Approach to Building Information Modeling : Integrated Practice Done the Right Way!* Second Edition ed: 4Site Press.

Khemlani, Lachmi. *Top Criteria for BIM Solutions: AECbytes Survey Results*. AECbytes, 10 de Outubro de 2007 2007 [cited 31-05.2013. Available from http://www.aecbytes.com/feature/2007/BIMSurveyReport.html.

Kiviniemi, Arto, Väino Tarandi;, Jan Karlshøj;, Håvard Bell;, and Ole Jørgen Karud;. 2008. "Review of the Development and Implementation of IFC compatible BIM." *Erabuild*.

Kymmell, W. 2008. *Building Information Modeling: Planning and Managing Construction Projects with 4D CAD and Simulations*: McGraw-Hill Professional Publishing.

Laakso, Mikael, and Arto Kiviniemi. 2012. "The IFC Standard - A Review of History, Development and Standardization." *ITcon - Jornal of Information Technology in Construction*:28.

Lehrenfeld, Georg, Wolfgang Mueller, and Norbert Wiechers. 1993. Conformance tests of very large STEP files. edited by Paderborn University. Warburger Str. 100, D-33098 Paderborn, Germany.

Liebich, Thomas. 2013. IFC4 the new buildingSMART Standard.

Lillja, Christopher M. *How Building Information Modeling (BIM) Improves the Design and Construction Process* 2012. Available from http://news.princeton.edu.

Martins, Maria Peixoto; João Poças. 2012. Pesquisa estruturada e manipulação de informação no modelo IFC. Requisitos e soluções. In *Congresso Construção 2012*. Coimbra, Portugal.

Nour, M. 2008. *Construction Product Data and Building Information Models*: VDM Publishing.

Pazlar, Tomaz, and Ziga Turk. 2008. "Interoperability in practice: Geometric data exchange using the IFC standard." *Electronic Journal of Information Technology in Construction* no. 13:362-380.

Schevers, Hans, and Robin Drogemuller. 2006. Converting the Industry Foundation Classes to the Web Ontology Language. edited by CSIRO Manufacturing & Information Technology. 37 Graham Road, Highett, Victoria 3190, Australia: IEEE Computer Society.

SCRA. 2006. STEP Application Handbook. *ISO 10303* Version 3: 181.

Silva, Victor. 2011. BIM - The summary of a long history.

Sinclair, Dale. 2012. BIM Overlay to the RIBA Outline Plan Work. In *Royal Institute of British Architects*: RIBA Publishing.

Thein, Volker. 2011. BIM Interoperability Through a Vendor-Independent File Format. *Industry Foundation Classes (IFC)*: 8.

Venugopal, Manu. 2011. *Formal Specification of Industry Foundation Class Concepts Using Engineering Ontologies*, Georgia Institute of Technology.

YIN, Robert K. 1993. *Applications of case study research.* Thousand Oaks, California: Sage Publications.

NORMAS

ISO 639 – *Codes for the representation of names of languages.*

ISO 8601 – *Representation of dates and times.*

ISO 10303 – *Automation Systems and Integration.*

ISO 10303-11:2004 – *Industrial automation systems and integration – Product data representation and exchange – Part 11: Description methods: The EXPRESS language reference manual.*

ISO 10303-21:2002 – *Industrial automation systems and integration – Product data representation and exchange – Part 21: Implementation methods: Clear text encoding of the exchange structure.*

ISO 10303-28:2007 – *Industrial automation systems and integration – Product data representation and exchange – Part 28: Implementation methods: XML representations of EXPRESS schemas and data, using XML schemas.*

ISO 12006-3:2007 – *Building construction – Organization of information about construction works – Part 3: Framework for object-oriented information.*

ISO/PAS 16739:2005 – *Industry Foundation Classes, Release 2x, Platform Specification (IFC2x Platform.*

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

ISO/IEC 11578:1996 – *Information technology – Open Systems Interconnection – Remote Procedure Call (RPC).*

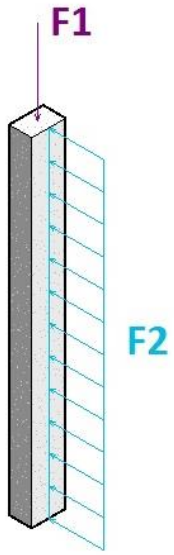
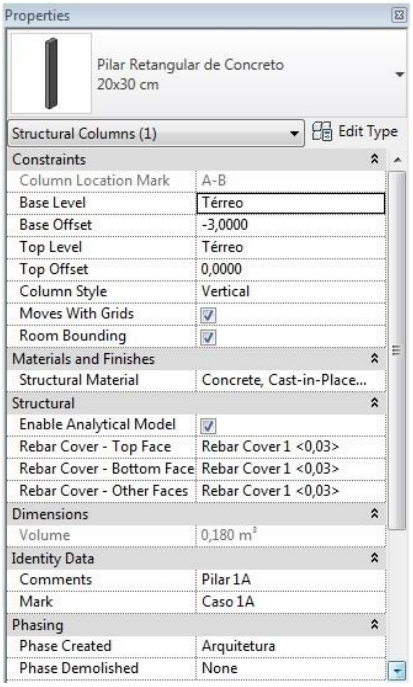
ISO 29481-2:2012 – *Building information models – Information delivery manual – Part 2: Interaction framework.*

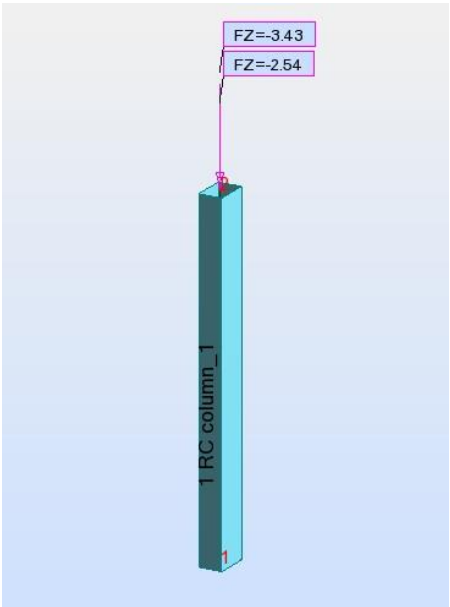
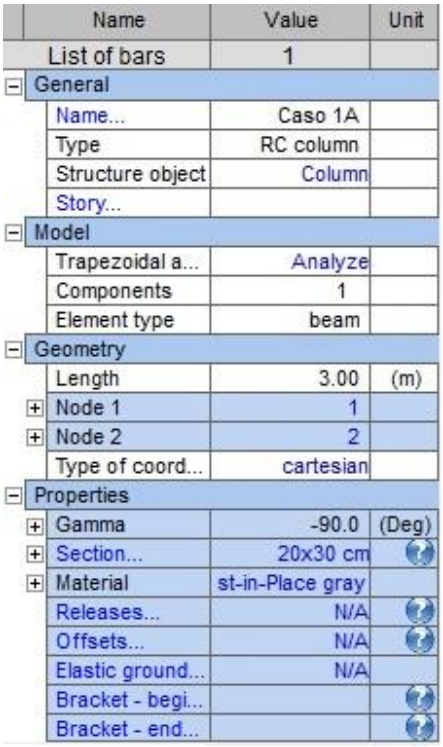
ANEXOS

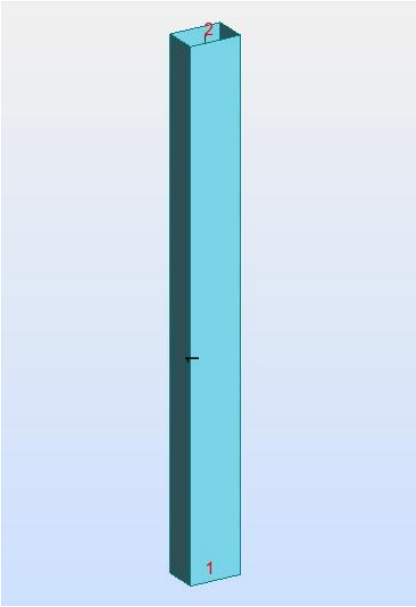
A1

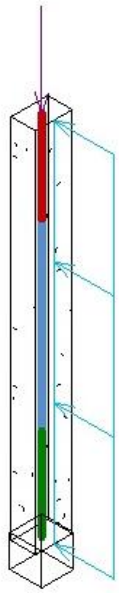
REPRESENTAÇÃO DE UM PILAR NOS VÁRIOS CASOS DE INTEROPERABILIDADE

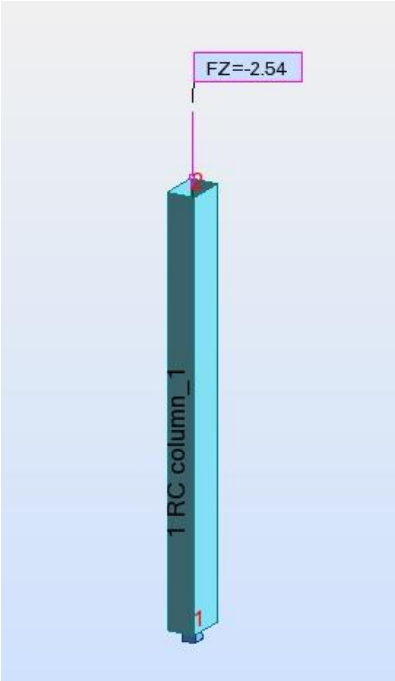
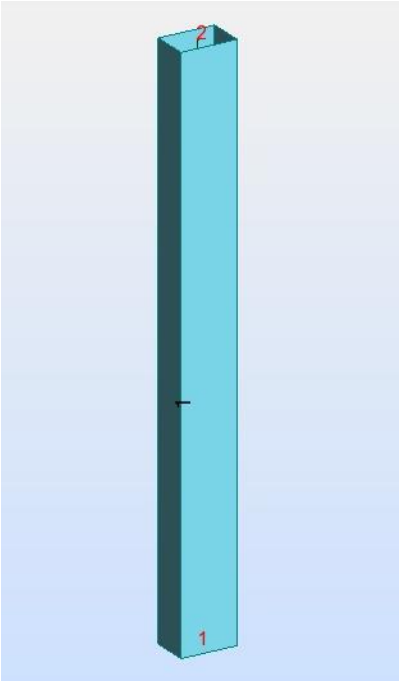
<table>
<tr><td colspan="3">Caso 1A</td></tr>
<tr>
<td>Revit Architecture

(Informação na Origem)</td>
<td> 
</td>
<td> 
</td>
</tr>
<tr>
<td>Robot

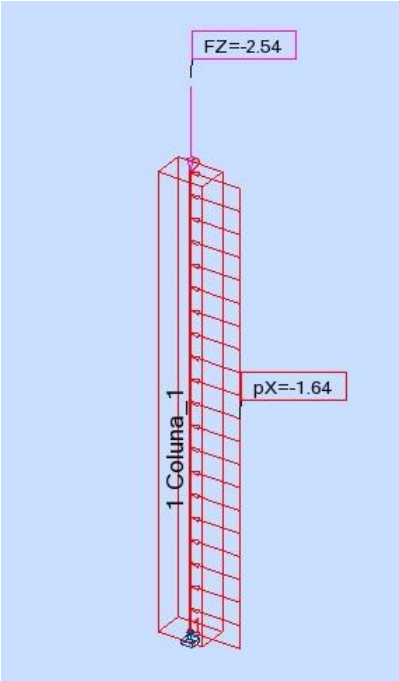
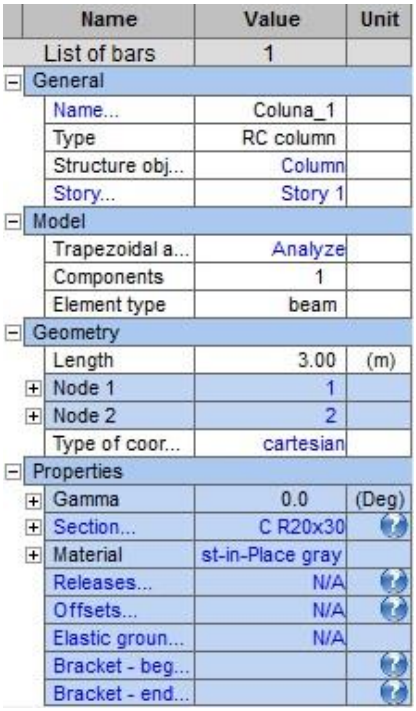
(Via Directa)</td>
<td> 
</td>
<td> 
</td>
</tr>
</table>

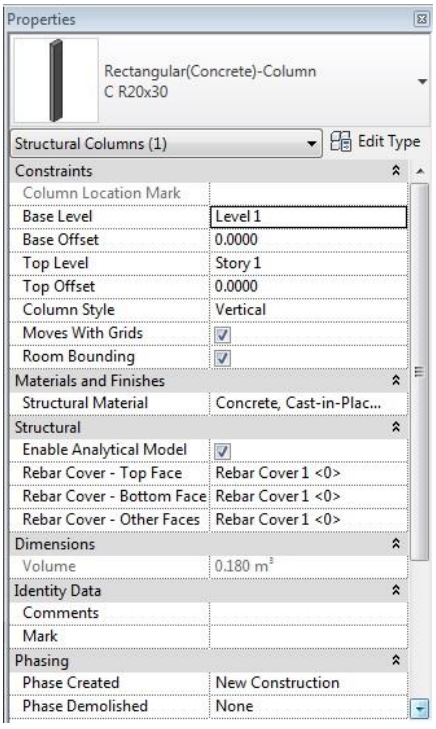
<table>
<tr>
<td>Robot
(Via IFC)</td>
<td> 
</td>
<td>
<table>
<tr><th>Name</th><th>Value</th><th>Unit</th></tr>
<tr><td colspan="2">List of bars</td><td>1</td></tr>
<tr><td colspan="3">General</td></tr>
<tr><td></td><td>Name...</td><td></td></tr>
<tr><td></td><td>Type</td><td>N/A</td></tr>
<tr><td></td><td>Structure object</td><td>Bar</td></tr>
<tr><td></td><td>Story...</td><td></td></tr>
<tr><td colspan="3">Model</td></tr>
<tr><td></td><td>Trapezoidal a...</td><td>Analyze</td></tr>
<tr><td></td><td>Components</td><td>1</td></tr>
<tr><td></td><td>Element type</td><td>beam</td></tr>
<tr><td colspan="3">Geometry</td></tr>
<tr><td></td><td>Length</td><td>3.00</td><td>(m)</td></tr>
<tr><td></td><td>Node 1</td><td>1</td></tr>
<tr><td></td><td>Node 2</td><td>2</td></tr>
<tr><td></td><td>Type of coord...</td><td>cartesian</td></tr>
<tr><td colspan="3">Properties</td></tr>
<tr><td></td><td>Gamma</td><td>0.0</td><td>(Deg)</td></tr>
<tr><td></td><td>Section...</td><td>IFC_R_20x30</td></tr>
<tr><td></td><td>Material</td><td>BETÃO</td></tr>
<tr><td></td><td>Releases...</td><td>N/A</td></tr>
<tr><td></td><td>Offsets...</td><td>N/A</td></tr>
<tr><td></td><td>Elastic ground...</td><td>N/A</td></tr>
<tr><td></td><td>Bracket - begi...</td></tr>
<tr><td></td><td>Bracket - end...</td></tr>
</table>
</td>
</tr>
</table>

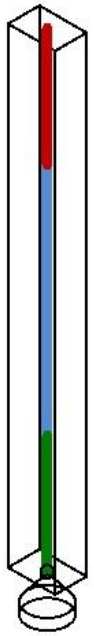
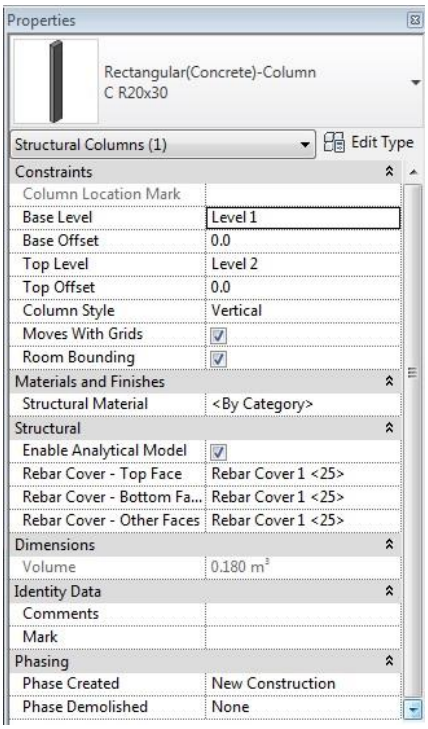
<table>
<tr><td colspan="3">Caso 1B</td></tr>
<tr>
<td>Revit Structures
(Informação na Origem)</td>
<td>  </td>
<td>
<table>
<tr><td colspan="2">Properties</td></tr>
<tr><td colspan="2">Concrete-Rectangular-Column
200 x 300mm</td></tr>
<tr><td colspan="2">Structural Columns (1) Edit Type</td></tr>
<tr><td colspan="2">Constraints</td></tr>
<tr><td>Column Location Mark</td><td>1-3</td></tr>
<tr><td>Base Level</td><td>Level 1</td></tr>
<tr><td>Base Offset</td><td>0.0</td></tr>
<tr><td>Top Level</td><td>Level 2</td></tr>
<tr><td>Top Offset</td><td>0.0</td></tr>
<tr><td>Column Style</td><td>Vertical</td></tr>
<tr><td>Moves With Grids</td><td>☑</td></tr>
<tr><td>Room Bounding</td><td>☑</td></tr>
<tr><td colspan="2">Materials and Finishes</td></tr>
<tr><td>Structural Material</td><td>Concrete, Cast-in-Place...</td></tr>
<tr><td colspan="2">Structural</td></tr>
<tr><td>Enable Analytical Model</td><td>☑</td></tr>
<tr><td>Rebar Cover - Top Face</td><td>Rebar Cover 1 <25></td></tr>
<tr><td>Rebar Cover - Bottom Face</td><td>Rebar Cover 1 <25></td></tr>
<tr><td>Rebar Cover - Other Faces</td><td>Rebar Cover 1 <25></td></tr>
<tr><td colspan="2">Dimensions</td></tr>
<tr><td>Volume</td><td>0.180 m³</td></tr>
<tr><td colspan="2">Identity Data</td></tr>
<tr><td>Comments</td><td>Pilar 1B</td></tr>
<tr><td>Mark</td><td>Caso 1B</td></tr>
<tr><td colspan="2">Phasing</td></tr>
<tr><td>Phase Created</td><td>New Construction</td></tr>
<tr><td>Phase Demolished</td><td>None</td></tr>
</table>
</td>
</tr>
</table>

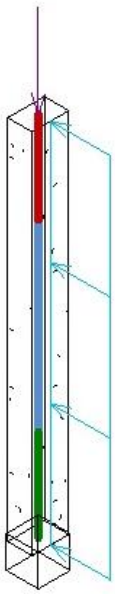
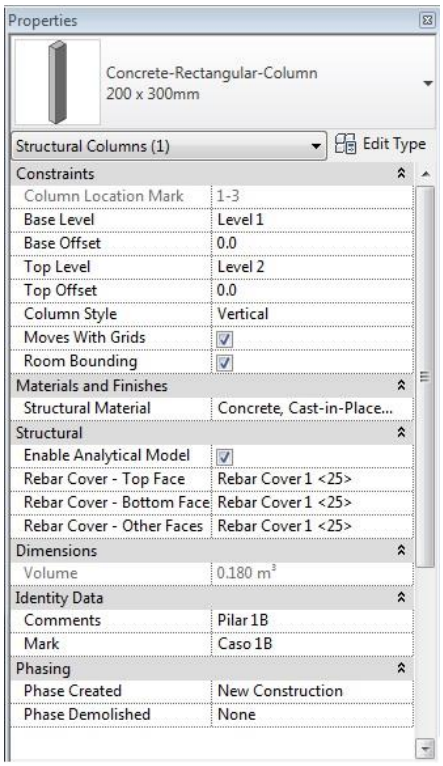
<table>
<tr>
<td>Robot
(Via Directa)</td>
<td> 
</td>
<td>
<table>
<tr><th></th><th>Name</th><th>Value</th><th>Unit</th></tr>
<tr><td colspan="2">List of bars</td><td>1</td><td></td></tr>
<tr><td colspan="4">General</td></tr>
<tr><td></td><td>Name...</td><td>Caso 1B</td><td></td></tr>
<tr><td></td><td>Type</td><td>RC column</td><td></td></tr>
<tr><td></td><td>Structure...</td><td>Column</td><td></td></tr>
<tr><td></td><td>Story...</td><td></td><td></td></tr>
<tr><td colspan="4">Model</td></tr>
<tr><td></td><td>Trapezoi...</td><td>Analyze</td><td></td></tr>
<tr><td></td><td>Compone...</td><td>1</td><td></td></tr>
<tr><td></td><td>Element t...</td><td>beam</td><td></td></tr>
<tr><td colspan="4">Geometry</td></tr>
<tr><td></td><td>Length</td><td>3.00</td><td>(m)</td></tr>
<tr><td></td><td>Node 1</td><td>1</td><td></td></tr>
<tr><td></td><td>Node 2</td><td>2</td><td></td></tr>
<tr><td></td><td>Type of c...</td><td>cartesian</td><td></td></tr>
<tr><td colspan="4">Properties</td></tr>
<tr><td></td><td>Gamma</td><td>-90.0</td><td>(Deg)</td></tr>
<tr><td></td><td>Section...</td><td>200 x 300mm</td><td></td></tr>
<tr><td></td><td>Material</td><td>st-in-Place gray</td><td></td></tr>
<tr><td></td><td>Releases...</td><td>N/A</td><td></td></tr>
<tr><td></td><td>Offsets...</td><td>N/A</td><td></td></tr>
<tr><td></td><td>Elastic gr...</td><td>N/A</td><td></td></tr>
<tr><td></td><td>Bracket - ...</td><td></td><td></td></tr>
<tr><td></td><td>Bracket - ...</td><td></td><td></td></tr>
</table>
</td>
</tr>
<tr>
<td>Robot
(Via IFC)</td>
<td> 
</td>
<td>
<table>
<tr><th></th><th>Name</th><th>Value</th><th>Unit</th></tr>
<tr><td colspan="2">List of bars</td><td>1</td><td></td></tr>
<tr><td colspan="4">General</td></tr>
<tr><td></td><td>Name...</td><td></td><td></td></tr>
<tr><td></td><td>Type</td><td>N/A</td><td></td></tr>
<tr><td></td><td>Structure o...</td><td>Bar</td><td></td></tr>
<tr><td></td><td>Story...</td><td></td><td></td></tr>
<tr><td colspan="4">Model</td></tr>
<tr><td></td><td>Trapezoida...</td><td>Analyze</td><td></td></tr>
<tr><td></td><td>Components</td><td>1</td><td></td></tr>
<tr><td></td><td>Element type</td><td>beam</td><td></td></tr>
<tr><td colspan="4">Geometry</td></tr>
<tr><td></td><td>Length</td><td>3.00</td><td>(m)</td></tr>
<tr><td></td><td>Node 1</td><td>1</td><td></td></tr>
<tr><td></td><td>Node 2</td><td>2</td><td></td></tr>
<tr><td></td><td>Type of co...</td><td>cartesian</td><td></td></tr>
<tr><td colspan="4">Properties</td></tr>
<tr><td></td><td>Gamma</td><td>0.0</td><td>(Deg)</td></tr>
<tr><td></td><td>Section...</td><td>IFC_R_20x30</td><td></td></tr>
<tr><td></td><td>Material</td><td>BETÃO</td><td></td></tr>
<tr><td></td><td>Releases...</td><td>N/A</td><td></td></tr>
<tr><td></td><td>Offsets...</td><td>N/A</td><td></td></tr>
<tr><td></td><td>Elastic gro...</td><td>N/A</td><td></td></tr>
<tr><td></td><td>Bracket - b...</td><td></td><td></td></tr>
<tr><td></td><td>Bracket - e...</td><td></td><td></td></tr>
</table>
</td>
</tr>
</table>

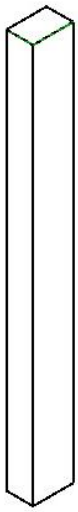
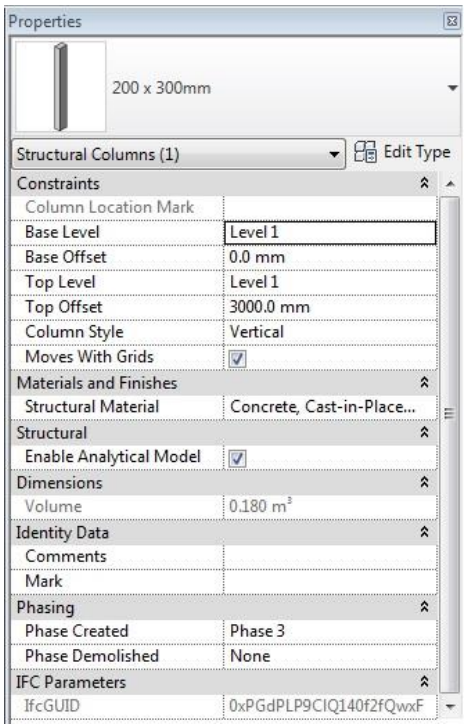
<table>
<tr><td colspan="3">Caso 1C e 1D</td></tr>
<tr>
<td>Robot

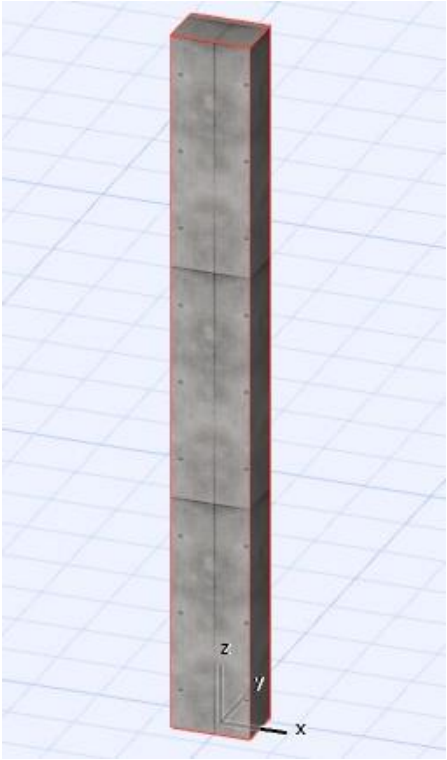
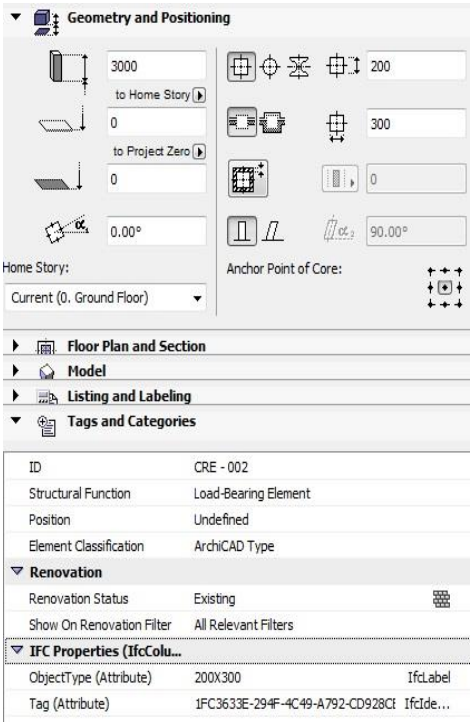
(Informação na Origem)</td>
<td> 
</td>
<td> 
</td>
</tr>
<tr>
<td>Revit Architecture

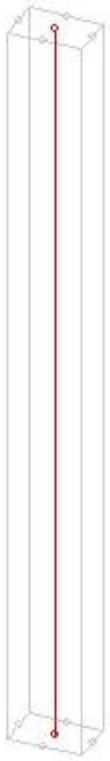
(Via Directa)</td>
<td> </td>
<td> 
</td>
</tr>
</table>

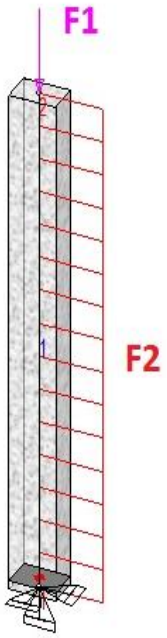
Revit Structures (Via Directa)		

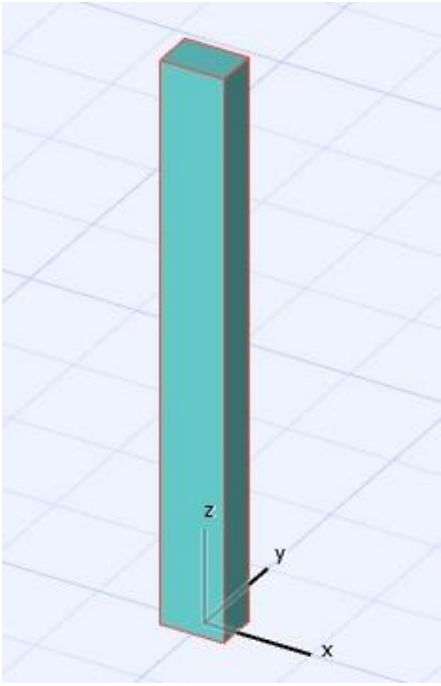
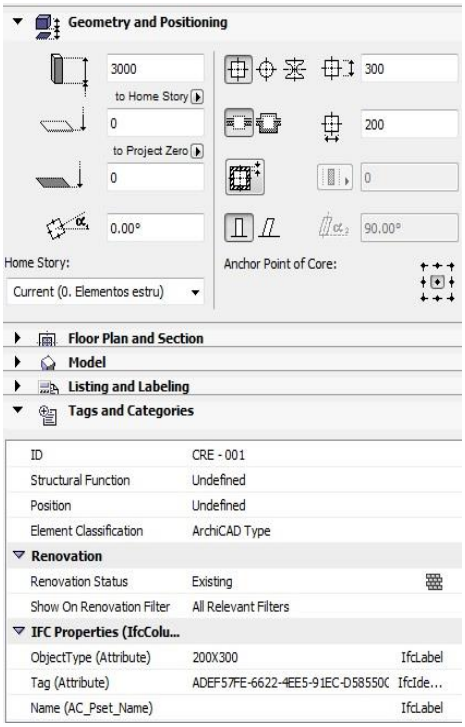
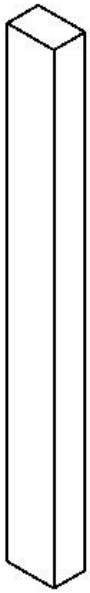
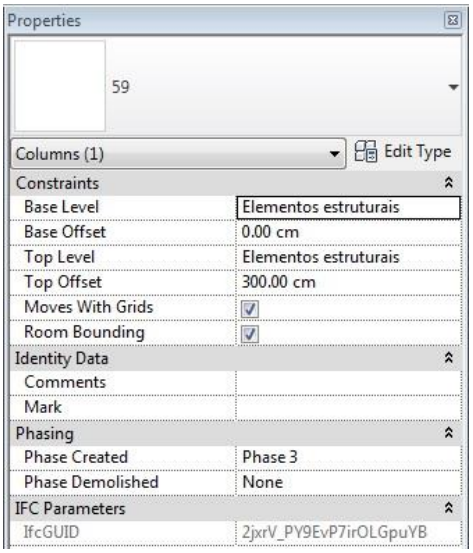
Caso 1E		
Revit Structures (Informação na Origem)		

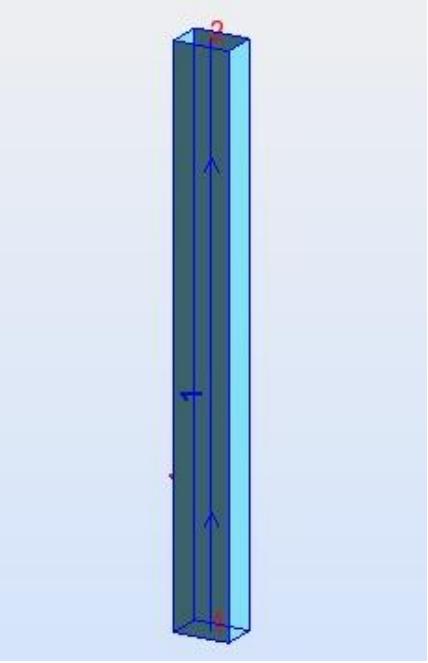
Revit Structures (Via IFC)		

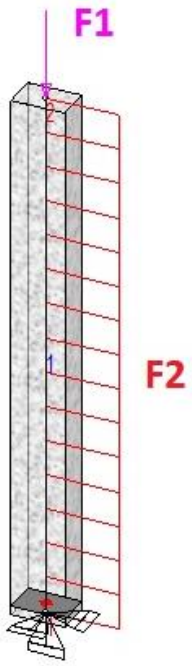
Caso 1F		
ArchiCad (Informação na Origem)		

Tricalc (Via IFC)		No Tricalc as propriedades podem ser consultadas através de listagens e relatórios completos gerados em ficheiro do tipo “*.docx”.

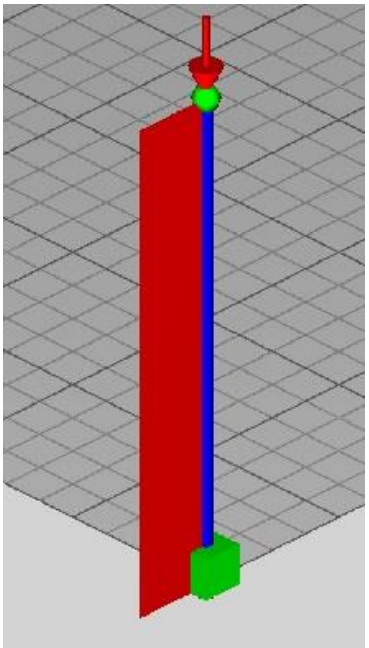
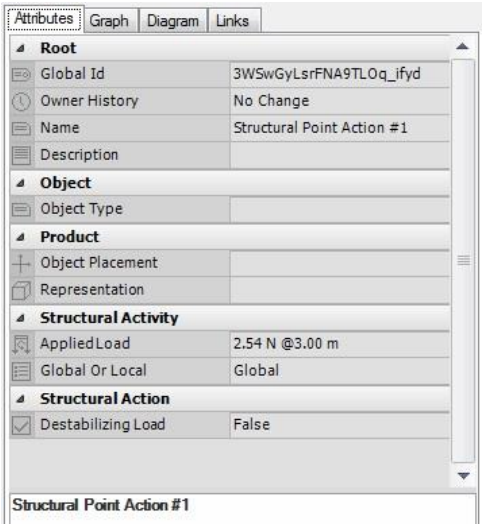
Caso 1G		
Tricalc (Informação na Origem)		No Tricalc as propriedades podem ser consultadas através de listagens e relatórios completos gerados em ficheiro do tipo “*.docx”.

Archicad (Via IFC)		
Revit Structures (Via IFC)		

<table>
<tr>
<td>Robot
(Via IFC)</td>
<td>  </td>
<td>
<table>
<tr><th></th><th>Nome</th><th>Valor</th><th>Unida</th></tr>
<tr><td colspan="2">Lista de barras</td><td>1</td><td></td></tr>
<tr><td colspan="4">Geral</td></tr>
<tr><td></td><td>Nome...</td><td></td><td></td></tr>
<tr><td></td><td>Tipo</td><td>N/A</td><td></td></tr>
<tr><td></td><td>Objeto estr...</td><td>Barra</td><td></td></tr>
<tr><td></td><td>Andar...</td><td></td><td></td></tr>
<tr><td colspan="4">Modelo</td></tr>
<tr><td></td><td>Método tra...</td><td>Analisar</td><td></td></tr>
<tr><td></td><td>Component...</td><td>1</td><td></td></tr>
<tr><td></td><td>Tipo de ele...</td><td>viga</td><td></td></tr>
<tr><td colspan="4">Geometria</td></tr>
<tr><td></td><td>Comprimento</td><td>3.00</td><td>(m)</td></tr>
<tr><td>+</td><td>Nó 1</td><td>1</td><td></td></tr>
<tr><td>+</td><td>Nó 2</td><td>2</td><td></td></tr>
<tr><td></td><td>Tipo de sist...</td><td>cartesiano</td><td></td></tr>
<tr><td colspan="4">Propriedades</td></tr>
<tr><td>+</td><td>Gama</td><td>0.0</td><td>(Deg)</td></tr>
<tr><td>+</td><td>Seção...</td><td>IFC_R_20x30</td><td></td></tr>
<tr><td>+</td><td>Material</td><td>BETÃO</td><td></td></tr>
<tr><td></td><td>Liberações...</td><td>N/A</td><td></td></tr>
<tr><td></td><td>Deslocame...</td><td>N/A</td><td></td></tr>
<tr><td></td><td>Solo elástic...</td><td>N/A</td><td></td></tr>
<tr><td></td><td>Reforço - i...</td><td></td><td></td></tr>
<tr><td></td><td>Reforço - fi...</td><td></td><td></td></tr>
</table>
</td>
</tr>
</table>

<table>
<tr><td colspan="3">Caso 1H</td></tr>
<tr>
<td>Tricalc
(Informação na Origem)</td>
<td> 
</td>
<td>No Tricalc as propriedades podem ser consultadas através de listagens e relatórios completos gerados em ficheiro do tipo “*.docx”.</td>
</tr>
</table>

Tricalc (Via IFC)		No Tricalc as propriedades podem ser consultadas através de listagens e relatórios completos gerados em ficheiro do tipo "*.docx".

Caso 1I		
IFC (Visualizado com o *software* Constructivity) (Informação na Origem)		

A2

TEXTO DE CÓDIGO DO MODELO USADO NA ANÁLISE E EXPOSIÇÃO DE UM FICHEIRO IFC

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION (('ViewDefinition [CoordinationView]'),'2;1');
FILE_NAME ('ModeloIFC.ifc','2013-04-03T10:54:01','Sérgio
Pinho','FEUP','BIM','Notepad++',$);
FILE_SCHEMA (('IFC4'));
ENDSEC;

DATA;
/* Dados do usuário, organização e da aplicação */
#1= IFCORGANIZATION ($,'Sergio-PC','FEUP',$,$);
#2= IFCAPPLICATION (#1,'1.0','IFC4','Made via Notepad++');
#3= IFCPERSON ($,$,'Sergio',$,$,$,$,$);
#4= IFCPERSONANDORGANIZATION (#3,#1,$);
#5= IFCOWNERHISTORY (#4,#2,.READWRITE.,.NOCHANGE.,$,$,$,1320677205);


/* Pontos cartesianos e direcções */
#7= IFCCARTESIANPOINT ((0.,0.,0.));
#8= IFCCARTESIANPOINT ((0.,0.));
#9= IFCDIRECTION ((1.,0.,0.));
#10= IFCDIRECTION ((-1.,0.,0.));
#11= IFCDIRECTION ((0.,1.,0.));
#12= IFCDIRECTION ((0.,-1.,0.));
#13= IFCDIRECTION ((0.,0.,1.));
#14= IFCDIRECTION ((0.,0.,-1.));
#15= IFCDIRECTION ((1.,0.));
#16= IFCDIRECTION ((-1.,0.));
#17= IFCDIRECTION ((0.,1.));
#18= IFCDIRECTION ((0.,-1.));


/* Unidades */
#20= IFCSIUNIT (*,.AREAUNIT.,$,.SQUARE_METRE.);
#21= IFCSIUNIT (*,.FORCEUNIT.,$,.NEWTON.);
#28= IFCSIUNIT (*,.LENGTHUNIT.,$,.METRE.);
#36= IFCSIUNIT (*,.MASSUNIT.,.KILO.,.GRAM.);
#40= IFCSIUNIT (*,.PLANEANGLEUNIT.,$,.RADIAN.);
#41= IFCMEASUREWITHUNIT (IFCPLANEANGLEMEASURE(0.0174532925199433),#40);
#42= IFCDIMENSIONALEXPONENTS (0,0,0,0,0,0,0);
#43= IFCCONVERSIONBASEDUNIT (#42,.PLANEANGLEUNIT.,'DEGREE',#41);
#45= IFCSIUNIT (*,.PRESSUREUNIT.,$,.PASCAL.);
#56= IFCSIUNIT (*,.VOLUMEUNIT.,$,.CUBIC_METRE.);
#57= IFCSIUNIT (*,.TIMEUNIT.,$,.SECOND.);
#58= IFCSIUNIT (*,.THERMODYNAMICTEMPERATUREUNIT.,$,.DEGREE_CELSIUS.);
#59= IFCSIUNIT (*,.LUMINOUSINTENSITYUNIT.,$,.LUMEN.);
#96= IFCDERIVEDUNITELEMENT (#21,1);
#97= IFCDERIVEDUNITELEMENT (#28,-1);
#98= IFCDERIVEDUNIT ((#96,#97),.LINEARFORCEUNIT.,$);
#99= IFCDERIVEDUNITELEMENT (#21,1);
#100= IFCDERIVEDUNITELEMENT (#28,1);
#101= IFCDERIVEDUNITELEMENT (#28,-1);
#102= IFCDERIVEDUNIT ((#99,#100,#101),.LINEARMOMENTUNIT.,$);
#103= IFCDERIVEDUNITELEMENT (#21,1);
#104= IFCDERIVEDUNITELEMENT (#28,-1);
#105= IFCDERIVEDUNIT ((#103,#104),.LINEARSTIFFNESSUNIT.,$);
#112= IFCDERIVEDUNITELEMENT (#36,1);
#113= IFCDERIVEDUNITELEMENT (#56,-1);
#114= IFCDERIVEDUNIT ((#112,#113),.MASSDENSITYUNIT.,$);
#118= IFCDERIVEDUNITELEMENT (#36,1);
#119= IFCDERIVEDUNITELEMENT (#28,-1);
#120= IFCDERIVEDUNIT ((#118,#119),.MASSPERLENGTHUNIT.,$);
#121= IFCDERIVEDUNITELEMENT (#45,1);
#122= IFCDERIVEDUNIT ((#121),.MODULUSOFELASTICITYUNIT.,$);
#140= IFCDERIVEDUNITELEMENT (#28,4);
```

```
#141= IFCDERIVEDUNIT ((#140),.MOMENTOFINERTIAUNIT.,$);
#142= IFCDERIVEDUNITELEMENT (#21,1);
#143= IFCDERIVEDUNITELEMENT (#20,-1);
#144= IFCDERIVEDUNIT ((#142,#143),.PLANARFORCEUNIT.,$);
#147= IFCDERIVEDUNITELEMENT (#36,1);
#148= IFCDERIVEDUNITELEMENT (#20,-1);
#149= IFCDERIVEDUNIT ((#147,#148),.ROTATIONALMASSUNIT.,$);
#150= IFCDERIVEDUNITELEMENT (#21,1);
#151= IFCDERIVEDUNITELEMENT (#28,1);
#152= IFCDERIVEDUNITELEMENT (#43,-1);
#153= IFCDERIVEDUNIT ((#150,#151,#152),.ROTATIONALSTIFFNESSUNIT.,$);
#154= IFCDERIVEDUNITELEMENT (#28,5);
#155= IFCDERIVEDUNIT ((#154),.SECTIONAREAINTEGRALUNIT.,$);
#156= IFCDERIVEDUNITELEMENT (#28,3);
#157= IFCDERIVEDUNIT ((#156),.SECTIONMODULUSUNIT.,$);
#158= IFCDERIVEDUNITELEMENT (#45,1);
#159= IFCDERIVEDUNIT ((#158),.SHEARMODULUSUNIT.,$);
#160= IFCUNITASSIGNMENT
((#20,#21,#28,#36,#43,#45,#56,#98,#102,#105,#114,#120,#122,#141,#144,#149,#153,#155
,#157,#159));


/* Projecto */
#180= IFCPROJECT ('0QjBRF7yLCTh4HRF3GOF1t',#5,'Project',$,$,$,$,(#202,#203),#160);


/* Localização do Projecto */
#190= IFCAXIS2PLACEMENT2D (#8,#15);
#191= IFCAXIS2PLACEMENT3D (#7,$,$);
#192= IFCLOCALPLACEMENT ($,#191);
#193= IFCBUILDING ('2QhGKmE2bDmA5FWgSPSo81',#5,$,$,$,#192,$,$,.ELEMENT.,$,$,#194);
#194= IFCPOSTALADDRESS (.HOME.,$,$,'Arões','Rua da Aldeia','n°93','Vale de
Cambra','Aveiro','3730-001','Portugal');
#195= IFCBUILDINGSTOREY ('1QhGKmE2bDmA5FWgVcZDp7',#5,'Ground',$,$,#192,$,'Rés-do-
Chão',.COMPLEX.,0.);
#196= IFCBUILDINGSTOREY
('2QhGKmE2bDmA5FWgVcZDp7',#5,'Floor',$,$,#192,$,'1°Piso',.COMPLEX.,3.);
#197= IFCSITE
('2EfWEdX2bDmQ3GVgSPSo82',#5,'Site',$,$,#192,$,$,.ELEMENT.,(40,47,57),(8,17,54),634
.,$,$);
#198= IFCRELAGGREGATES ('8DrHRdX5lONR8JKpYUIp75',#5,$,$,#197,#193);
#199= IFCRELAGGREGATES ('9SpBQdB63LOK4FGpDPAp32',#5,$,$,#193,(#195,#196));

/* Representação geométrica */
#202= IFCGEOMETRICREPRESENTATIONCONTEXT ('3D','Model',3,0.00001,#191,$);
#203= IFCGEOMETRICREPRESENTATIONCONTEXT ('2D','Plan',2,0.00001,#191,$);
#204= IFCGEOMETRICREPRESENTATIONSUBCONTEXT ('Body','Model',#202,$,.MODEL_VIEW.,$);
#205= IFCGEOMETRICREPRESENTATIONSUBCONTEXT
('Annotation','Plan',#203,$,.PLAN_VIEW.,$);

/* ---------------------------------------------------------------------------- */


/* Pilar 1 - Nó inferior */
#220= IFCCARTESIANPOINT ((0.,0.,0.));
#221= IFCVERTEXPOINT (#220);
#222= IFCTOPOLOGYREPRESENTATION (#202,'Reference','Vertex',(#221));
#223= IFCPRODUCTDEFINITIONSHAPE ($,$,(#222));
#224= IFCSTRUCTURALPOINTCONNECTION ('1539fAVu96i8mFr0cgUqeI',#5,'Point Connection
1',$,$,$,#223,#225,$);

/* Pilar 1 - Restrições nó inferior - Fixo */
#225= IFCBOUNDARYNODECONDITION
('Fixed',IFCBOOLEAN(.T.),IFCBOOLEAN(.T.),IFCBOOLEAN(.T.),IFCBOOLEAN(.T.),IFCBOOLEAN
(.T.),IFCBOOLEAN(.T.));
```

```
/* Pilar 1 - Nó superior */
#226= IFCCARTESIANPOINT ((0.,0.,3.));
#227= IFCVERTEXPOINT (#226);
#228= IFCTOPOLOGYREPRESENTATION (#202,'Reference','Vertex',(#227));
#229= IFCPRODUCTDEFINITIONSHAPE ($,$,(#228));
#230= IFCSTRUCTURALPOINTCONNECTION ('1mc6ibF258HPIpTmqg6DSl',#5,'Point Connection
2',$,$,$,#229,#231,$);

/* Pilar 1 - Restrições nó superior - Livre */
#231= IFCBOUNDARYNODECONDITION
('Free',IFCBOOLEAN(.F.),IFCBOOLEAN(.F.),IFCBOOLEAN(.F.),IFCBOOLEAN(.F.),IFCBOOLEAN(
.F.),IFCBOOLEAN(.F.));

/* Pilar 2 - Nó inferior */
#240= IFCCARTESIANPOINT ((5.5,0.,0.));
#241= IFCVERTEXPOINT (#240);
#242= IFCTOPOLOGYREPRESENTATION (#202,'Reference','Vertex',(#241));
#243= IFCPRODUCTDEFINITIONSHAPE ($,$,(#242));
#244= IFCSTRUCTURALPOINTCONNECTION ('2539fAVu96i8mFr0cgUqeI',#5,'Point Connection
4',$,$,$,#243,#245,$);


/* Pilar 2 - Restrições nó inferior - Fixo */
#245= IFCBOUNDARYNODECONDITION
('Fixed',IFCBOOLEAN(.T.),IFCBOOLEAN(.T.),IFCBOOLEAN(.T.),IFCBOOLEAN(.T.),IFCBOOLEAN
(.T.),IFCBOOLEAN(.T.));

/* Pilar 2 - Nó superior */
#246= IFCCARTESIANPOINT ((5.5,0.,3.));
#247= IFCVERTEXPOINT (#246);
#248= IFCTOPOLOGYREPRESENTATION (#202,'Reference','Vertex',(#247));
#249= IFCPRODUCTDEFINITIONSHAPE ($,$,(#248));
#250= IFCSTRUCTURALPOINTCONNECTION ('2mc6ibF258HPIpTmqg6DSl',#5,'Point Connection
3',$,$,$,#249,#251,$);

/* Pilar 2 - Restrições nó superior - Livre */
#251= IFCBOUNDARYNODECONDITION
('Free',IFCBOOLEAN(.F.),IFCBOOLEAN(.F.),IFCBOOLEAN(.F.),IFCBOOLEAN(.F.),IFCBOOLEAN(
.F.),IFCBOOLEAN(.F.));

/* Pilar 3 - Nó Inferior */
#260= IFCCARTESIANPOINT ((0.,5.5,0.));
#261= IFCVERTEXPOINT (#260);
#262= IFCTOPOLOGYREPRESENTATION (#202,'Reference','Vertex',(#261));
#263= IFCPRODUCTDEFINITIONSHAPE ($,$,(#262));
#264= IFCSTRUCTURALPOINTCONNECTION ('3339fAVu96i8mFr0cgUqeI',#5,'Point Connection
5',$,$,$,#263,#265,$);

/* Pilar 3 - Restrições nó inferior - Fixo */
#265= IFCBOUNDARYNODECONDITION
('Fixed',IFCBOOLEAN(.T.),IFCBOOLEAN(.T.),IFCBOOLEAN(.T.),IFCBOOLEAN(.T.),IFCBOOLEAN
(.T.),IFCBOOLEAN(.T.));

/* Pilar 3 - Nó superior */
#266= IFCCARTESIANPOINT ((0.,5.5,3.));
#267= IFCVERTEXPOINT (#266);
#268= IFCTOPOLOGYREPRESENTATION (#202,'Reference','Vertex',(#267));
#269= IFCPRODUCTDEFINITIONSHAPE ($,$,(#268));
#270= IFCSTRUCTURALPOINTCONNECTION ('3mc6ibF258HPIpTmqg6DSl',#5,'Point Connection
6',$,$,$,#269,#271,$);

/* Pilar 3 - Restrições nó superior - Livre */
```

```
#271= IFCBOUNDARYNODECONDITION
('Free',IFCBOOLEAN(.F.),IFCBOOLEAN(.F.),IFCBOOLEAN(.F.),IFCBOOLEAN(.F.),IFCBOOLEAN(
.F.),IFCBOOLEAN(.F.));


/* Pilar 4 - Nó inferior */
#280= IFCCARTESIANPOINT ((5.5,5.5,0.));
#281= IFCVERTEXPOINT (#280);
#282= IFCTOPOLOGYREPRESENTATION (#202,'Reference','Vertex',(#281));
#283= IFCPRODUCTDEFINITIONSHAPE ($,$,(#282));
#284= IFCSTRUCTURALPOINTCONNECTION ('4539fAVu96i8mFr0cgUqeI',#5,'Point Connection
8',$,$,$,#283,#285,$);

/* Pilar 4 - Restrições nó inferior - Fixo */
#285= IFCBOUNDARYNODECONDITION
('Fixed',IFCBOOLEAN(.T.),IFCBOOLEAN(.T.),IFCBOOLEAN(.T.),IFCBOOLEAN(.T.),IFCBOOLEAN
(.T.),IFCBOOLEAN(.T.));

/* Pilar 4 - Nó superior */
#286= IFCCARTESIANPOINT ((5.5,5.5,3.));
#287= IFCVERTEXPOINT (#286);
#288= IFCTOPOLOGYREPRESENTATION (#202,'Reference','Vertex',(#287));
#289= IFCPRODUCTDEFINITIONSHAPE ($,$,(#288));
#290= IFCSTRUCTURALPOINTCONNECTION ('4mc6ibF258HPIpTmqg6DSl',#5,'Point Connection
7',$,$,$,#289,#291,$);

/* Pilar 4 - Restrições nó superior - Livre */
#291= IFCBOUNDARYNODECONDITION
('Free',IFCBOOLEAN(.F.),IFCBOOLEAN(.F.),IFCBOOLEAN(.F.),IFCBOOLEAN(.F.),IFCBOOLEAN(
.F.),IFCBOOLEAN(.F.));


/* Pilar 1 - Membro estrutural (Objecto) */
#300= IFCDIRECTION ((1.,0.,0.));
#301= IFCSTRUCTURALCURVEMEMBER ('3eXlZ8csrAvfIIXVwC_gVP',#5,'Curve Member
1A',$,$,$,#306,.RIGID_JOINED_MEMBER.,#300);
#302= IFCRELCONNECTSSTRUCTURALMEMBER
('1Z9w70JcDEUx6TnggLE2wU',#5,$,$,#301,#224,$,$,$,$);
#303= IFCRELCONNECTSSTRUCTURALMEMBER
('1GClK7cwT80xzpZuaAlGXp',#5,$,$,#301,#230,$,$,$,$);

/* Pilar 1 - Membro estrutural (Representação) */
#304= IFCEDGE (#221,#227);
#305= IFCTOPOLOGYREPRESENTATION (#202,'Reference','Edge',(#304));
#306= IFCPRODUCTDEFINITIONSHAPE ($,$,(#305));


/* Pilar 2 - Membro estrutural (Objecto) */
#310= IFCDIRECTION ((1.,0.,0.));
#311= IFCSTRUCTURALCURVEMEMBER ('3eXlZ8csrAvfIIXVwC_gVP',#5,'Curve Member
2A',$,$,$,#316,.RIGID_JOINED_MEMBER.,#310);
#312= IFCRELCONNECTSSTRUCTURALMEMBER
('2Z9w70JcDEUx6TnggLE2wU',#5,$,$,#311,#244,$,$,$,$);
#313= IFCRELCONNECTSSTRUCTURALMEMBER
('2GClK7cwT80xzpZuaAlGXp',#5,$,$,#311,#250,$,$,$,$);

/* Pilar 2 - Membro estrutural (Representação) */
#314= IFCEDGE (#241,#247);
#315= IFCTOPOLOGYREPRESENTATION (#202,'Reference','Edge',(#314));
#316= IFCPRODUCTDEFINITIONSHAPE ($,$,(#315));


/* Pilar 3 - Membro estrutural (Objecto) */
#320= IFCDIRECTION ((1.,0.,0.));
```

```
#321= IFCSTRUCTURALCURVEMEMBER ('3eXlZ8csrAvfIIXVwC_gVP',#5,'Curve Member
3A',$,$,$,#326,.RIGID_JOINED_MEMBER.,#320);
#322= IFCRELCONNECTSSTRUCTURALMEMBER
('3Z9w70JcDEUx6TnggLE2wU',#5,$,$,#321,#264,$,$,$,$);
#323= IFCRELCONNECTSSTRUCTURALMEMBER
('3GClK7cwT80xzpZuaAlGXp',#5,$,$,#321,#270,$,$,$,$);

/* Pilar 3 - Membro estrutural (Representação) */
#324= IFCEDGE (#261,#267);
#325= IFCTOPOLOGYREPRESENTATION (#202,'Reference','Edge',(#324));
#326= IFCPRODUCTDEFINITIONSHAPE ($,$,(#325));


/* Pilar 4 - Membro estrutural (Objecto) */
#330= IFCDIRECTION ((1.,0.,0.));
#331= IFCSTRUCTURALCURVEMEMBER ('3eXlZ8csrAvfIIXVwC_gVP',#5,'Curve Member
4A',$,$,$,#336,.RIGID_JOINED_MEMBER.,#330);
#332= IFCRELCONNECTSSTRUCTURALMEMBER
('4Z9w70JcDEUx6TnggLE2wU',#5,$,$,#331,#284,$,$,$,$);
#333= IFCRELCONNECTSSTRUCTURALMEMBER
('4GClK7cwT80xzpZuaAlGXp',#5,$,$,#331,#290,$,$,$,$);

/* Pilar 4 - Membro estrutural (Representação) */
#334= IFCEDGE (#281,#287);
#335= IFCTOPOLOGYREPRESENTATION (#202,'Reference','Edge',(#334));
#336= IFCPRODUCTDEFINITIONSHAPE ($,$,(#335));


/* Viga 1 - Membro estrutural (Objecto) */
#350= IFCDIRECTION ((0.,0.,1.));
#351= IFCSTRUCTURALCURVEMEMBER ('25vEW7EzrBTvz5cbNWzhP$',#5,'Curve Member
1B',$,$,$,#356,.RIGID_JOINED_MEMBER.,#350);
#352= IFCRELCONNECTSSTRUCTURALMEMBER
('1ZUyJTZMHEev9njAeNDQUT',#5,$,$,#351,#230,$,$,$,$);
#353= IFCRELCONNECTSSTRUCTURALMEMBER
('1Y3WZZzV16XQ$1zEZLWjJX',#5,$,$,#351,#250,$,$,$,$);

/* Viga 1 - Membro estrutural (Representação) */
#354= IFCEDGE (#227,#247);
#355= IFCTOPOLOGYREPRESENTATION (#202,'Reference','Edge',(#354));
#356= IFCPRODUCTDEFINITIONSHAPE ($,$,(#355));


/* Viga 2 - Membro estrutural (Objecto) */
#360= IFCDIRECTION ((0.,0.,1.));
#361= IFCSTRUCTURALCURVEMEMBER ('25vEW7EzrBTvz5cbNWzhP$',#5,'Curve Member
2B',$,$,$,#366,.RIGID_JOINED_MEMBER.,#360);
#362= IFCRELCONNECTSSTRUCTURALMEMBER
('2ZUyJTZMHEev9njAeNDQUT',#5,$,$,#361,#270,$,$,$,$);
#363= IFCRELCONNECTSSTRUCTURALMEMBER
('2Y3WZZzV16XQ$1wEZLWjJX',#5,$,$,#361,#290,$,$,$,$);

/* Viga 2 - Membro estrutural (Representação) */
#364= IFCEDGE (#267,#287);
#365= IFCTOPOLOGYREPRESENTATION (#202,'Reference','Edge',(#364));
#366= IFCPRODUCTDEFINITIONSHAPE ($,$,(#365));


/* Viga 3 - Membro estrutural (Objecto) */
#370= IFCDIRECTION ((1.,0.,0.));
#371= IFCSTRUCTURALCURVEMEMBER ('25vEW7EzrBTvz5cbNWzhP$',#5,'Curve Member
3B',$,$,$,#376,.RIGID_JOINED_MEMBER.,#370);
#372= IFCRELCONNECTSSTRUCTURALMEMBER
('3ZUyJTZMHEev9njAeNDQUT',#5,$,$,#371,#230,$,$,$,$);
```

```
#373= IFCRELCONNECTSSTRUCTURALMEMBER
('3Y3WZZzV16XQ$1wEZLWjJX',#5,$,$,#371,#270,$,$,$,$);

/* Viga 3 - Membro estrutural (Representação) */
#374= IFCEDGE (#227,#267);
#375= IFCTOPOLOGYREPRESENTATION (#202,'Reference','Edge',(#374));
#376= IFCPRODUCTDEFINITIONSHAPE ($,$,(#375));


/* Viga 4 - Membro estrutural (Objecto) */
#380= IFCDIRECTION ((1.,0.,0.));
#381= IFCSTRUCTURALCURVEMEMBER ('25vEW7EzrBTvz5cbNWzhP$',#5,'Curve Member
4B',$,$,$,#386,.RIGID_JOINED_MEMBER.,#380);
#382= IFCRELCONNECTSSTRUCTURALMEMBER
('4ZUyJTZMHEev9njAeNDQUT',#5,$,$,#381,#250,$,$,$,$);
#383= IFCRELCONNECTSSTRUCTURALMEMBER
('4Y3WZZzV16XQ$1wEZLWjJX',#5,$,$,#381,#290,$,$,$,$);

/* Viga 4 - Membro estrutural (Representação) */
#384= IFCEDGE (#247,#287);
#385= IFCTOPOLOGYREPRESENTATION (#202,'Reference','Edge',(#384));
#386= IFCPRODUCTDEFINITIONSHAPE ($,$,(#385));


/* Laje 1 - Membro estrutural (Objecto) */
#390= IFCSTRUCTURALSURFACEMEMBER ('50xQG7EzrTYvz5cbDFzhP$',#5,'Surface Member
1C',$,$,$,#400,.BENDING_ELEMENT.,0.22);
#391= IFCSTRUCTURALSURFACECONNECTION ('0mc6ibF258HPIpTmqg6DSl',#5,'Surface
Connection 1',$,$,$,#400,$,$);
#392= IFCRELCONNECTSSTRUCTURALMEMBER
('1ZUyJTZMHEev9njAeNDQUT',#5,$,$,#390,#230,$,$,$,$);
#393= IFCRELCONNECTSSTRUCTURALMEMBER
('2Y3WZZzV16XQ$1wEZLWjJX',#5,$,$,#390,#250,$,$,$,$);
#394= IFCRELCONNECTSSTRUCTURALMEMBER
('3ZUyJTZMHEev9njAeNDQUT',#5,$,$,#390,#270,$,$,$,$);
#395= IFCRELCONNECTSSTRUCTURALMEMBER
('4Y3WZZzV16XQ$1wEZLWjJX',#5,$,$,#390,#290,$,$,$,$);

/* Laje 1 - Membro estrutural (Representação) */
#396= IFCPOLYLOOP ((#226,#246,#286,#266));
#397= IFCFACEOUTERBOUND (#396,.T.);
#398= IFCFACESURFACE ((#397),$,.T.);
#399= IFCTOPOLOGYREPRESENTATION (#202,'Reference','Face',(#398));
#400= IFCPRODUCTDEFINITIONSHAPE ($,$,(#399));


/* Atribuição das nós e membros estruturais ao modelo de análise estrutural */
#401= IFCRELASSIGNSTOGROUP
('3mDDAcu390$A5orhhUv4qv',#5,$,$,(#224,#230,#301,#244,#250,#311,#264,#270,#321,#284
,#290,#331,#351,#361,#371,#381,#390),.PRODUCT.,#875);

/* ---------------------------------------------------------------------------- */


/* Sapatas - Secção */
#404= IFCRECTANGLEPROFILEDEF (.AREA.,'150x200',#190,1.5,2.0);
#405= IFCEXTRUDEDAREASOLID (#404,#191,#13,0.75);
#406= IFCSHAPEREPRESENTATION (#202,'Body','SweptSolid',(#405));
#407= IFCPRODUCTDEFINITIONSHAPE ($,$,(#406));
#408= IFCPRESENTATIONLAYERASSIGNMENT ('Footing_Profile_A','No
description',(#406),$);
#409= IFCCARTESIANPOINT ((0.0,0.0,-0.75));
#410= IFCAXIS2PLACEMENT3D (#409,$,$);
#411= IFCLOCALPLACEMENT (#192,#410);
```

```
#412= IFCFOOTING ('1we2ARoE50netW0RKs4eva',#5,'Sapata
1',$,$,#411,#407,$,.PAD_FOOTING.);
#413= IFCCARTESIANPOINT ((5.5,0.0,-0.75));
#414= IFCAXIS2PLACEMENT3D (#413,$,$);
#415= IFCLOCALPLACEMENT (#192,#414);
#416= IFCFOOTING ('2we2ARoE50netQ4FGs4drw',#5,'Sapata
2',$,$,#415,#407,$,.PAD_FOOTING.);
#417= IFCCARTESIANPOINT ((0.0,5.5,-0.75));
#418= IFCAXIS2PLACEMENT3D (#417,$,$);
#419= IFCLOCALPLACEMENT (#192,#418);
#420= IFCFOOTING ('3we2ARoE50netZ5KPs4ako',#5,'Sapata
3',$,$,#419,#407,$,.PAD_FOOTING.);
#421= IFCCARTESIANPOINT ((5.5,5.5,-0.75));
#422= IFCAXIS2PLACEMENT3D (#421,$,$);
#423= IFCLOCALPLACEMENT (#192,#422);
#424= IFCFOOTING ('4we2ARoE50netX9RTs4thg',#5,'Sapata
4',$,$,#423,#407,$,.PAD_FOOTING.);

/* Pilares - Secção e características  */
#425= IFCRECTANGLEPROFILEDEF (.AREA.,'20x30',$,0.20,0.30);
#426= IFCEXTRUDEDAREASOLID (#425,#191,#13,3.0);
#427= IFCSHAPEREPRESENTATION (#202,'Body','SweptSolid',(#426));
#428= IFCPRESENTATIONLAYERASSIGNMENT ('Columns Profile','No description',(#427),$);
#429= IFCPRODUCTDEFINITIONSHAPE ($,$,(#427));
#430= IFCPROPERTYSINGLEVALUE ('MassPerLength',$,IFCMASSPERLENGTHMEASURE(25),$);
#431= IFCPROPERTYSINGLEVALUE ('CrossSectionArea',$,IFCAREAMEASURE(0.06),$);
#432= IFCPROPERTYSINGLEVALUE
('MomentOfInertiaY',$,IFCMOMENTOFINERTIAMEASURE(0.0002),$);
#433= IFCPROPERTYSINGLEVALUE
('MomentOfInertiaZ',$,IFCMOMENTOFINERTIAMEASURE(0.00045),$);
#434= IFCPROPERTYSINGLEVALUE
('TorsionalSectionModulus',$,IFCSECTIONMODULUSMEASURE(0.622),$);
#435= IFCPROFILEPROPERTIES
('Pset_ProfileMechanical',$,(#430,#431,#432,#433,#434),#425);
#436= IFCCARTESIANPOINT ((0.0,0.0,0.0));
#437= IFCAXIS2PLACEMENT3D (#436,#13,#9);
#438= IFCLOCALPLACEMENT (#192,#437);
#439= IFCCOLUMN ('1fv4DZfY55exwX8QDy8xmw',#5,'Column 1',$,$,#438,#429,$,.COLUMN.);
#440= IFCRELCONNECTSSTRUCTURALELEMENT
('1pDDAft390$A5guwnUv4ac',#5,$,$,(#301),#439);
#441= IFCCARTESIANPOINT ((5.5,0.,0.));
#442= IFCAXIS2PLACEMENT3D (#441,#13,#9);
#443= IFCLOCALPLACEMENT (#192,#442);
#444= IFCCOLUMN ('2ui4PZfD35egqW2RSy4fvw',#5,'Column 2',$,$,#443,#429,$,.COLUMN.);
#445= IFCRELCONNECTSSTRUCTURALELEMENT
('2pDDAft390$A5guwnUv4ac',#5,$,$,(#311),#444);
#446= IFCCARTESIANPOINT ((0.,5.5,0.));
#447= IFCAXIS2PLACEMENT3D (#446,#13,#9);
#448= IFCLOCALPLACEMENT (#192,#447);
#449= IFCCOLUMN ('3ui4PZfD35egqW2RSy4fvw',#5,'Column 3',$,$,#448,#429,$,.COLUMN.);
#450= IFCRELCONNECTSSTRUCTURALELEMENT
('3pDDAft390$A5guwnUv4ac',#5,$,$,(#321),#449);
#451= IFCCARTESIANPOINT ((5.5,5.5,0.));
#452= IFCAXIS2PLACEMENT3D (#451,#13,#9);
#453= IFCLOCALPLACEMENT (#192,#452);
#454= IFCCOLUMN ('4ui4PZfD35egqW2RSy4fvw',#5,'Column 4',$,$,#453,#429,$,.COLUMN.);
#455= IFCRELCONNECTSSTRUCTURALELEMENT
('4pDDAft390$A5guwnUv4ac',#5,$,$,(#331),#454);
#456= IFCRELCONTAINEDINSPATIALSTRUCTURE
('15kVXT9hb7C80NqRflZZF6',#5,$,$,(#439,#444,#449,#454),#196);

/* Vigas - Secção e características - Direcção X  */
#476= IFCRECTANGLEPROFILEDEF (.AREA.,'20x40',#190,0.2,0.4);
#477= IFCPROPERTYSINGLEVALUE ('MassPerLength',$,IFCMASSPERLENGTHMEASURE(25),$);
#478= IFCPROPERTYSINGLEVALUE ('CrossSectionArea',$,IFCAREAMEASURE(0.08),$);
```

```
#479= IFCPROPERTYSINGLEVALUE
('MomentOfInertiaY',$,IFCMOMENTOFINERTIAMEASURE(0.00027),$);
#480= IFCPROPERTYSINGLEVALUE
('MomentOfInertiaZ',$,IFCMOMENTOFINERTIAMEASURE(0.00107),$);
#481= IFCPROPERTYSINGLEVALUE
('TorsionalSectionModulus',$,IFCSECTIONMODULUSMEASURE(0.622),$);
#482= IFCPROFILEPROPERTIES
('Pset_ProfileMechanical',$,(#477,#478,#479,#480,#481),#476);
#483= IFCEXTRUDEDAREASOLID (#476,#191,#13,5.2);
#484= IFCSHAPEREPRESENTATION (#202,'Body','SweptSolid',(#483));
#485= IFCPRESENTATIONLAYERASSIGNMENT ('Beam Profile','No description',(#484),$);
#486= IFCPRODUCTDEFINITIONSHAPE ($,$,(#484));
#487= IFCCARTESIANPOINT ((0.0,0.15,2.8));
#488= IFCAXIS2PLACEMENT3D (#487,#11,#9);
#489= IFCLOCALPLACEMENT (#192,#488);
#490= IFCBEAM ('1zt8DTaE82lkeQ8FPj5gye',#5,'Beam 1',$,$,#489,#486,$,.BEAM.);
#491= IFCRELCONNECTSSTRUCTURALELEMENT
('1aDDAft390$A5guwnUv4xc',#5,$,$,(#351),#490);
#492= IFCCARTESIANPOINT ((5.5,0.15,2.8));
#493= IFCAXIS2PLACEMENT3D (#492,#11,#9);
#494= IFCLOCALPLACEMENT (#192,#493);
#495= IFCBEAM ('2zt8DTaE82lkeQ8FPj5gye',#5,'Beam 2',$,$,#494,#486,$,.BEAM.);
#496= IFCRELCONNECTSSTRUCTURALELEMENT
('2aDDAft390$A5guwnUv4xc',#5,$,$,(#361),#495);
#497= IFCRELCONTAINEDINSPATIALSTRUCTURE
('25kVXT9hb7C80NqRflZZF6',#5,$,$,(#490,#495),#196);

/* Vigas - Secção e características - Direcção Y */
#526= IFCRECTANGLEPROFILEDEF (.AREA.,'20x30',#190,0.2,0.4);
#527= IFCPROPERTYSINGLEVALUE ('MassPerLength',$,IFCMASSPERLENGTHMEASURE(25),$);
#528= IFCPROPERTYSINGLEVALUE ('CrossSectionArea',$,IFCAREAMEASURE(0.06),$);
#529= IFCPROPERTYSINGLEVALUE
('MomentOfInertiaY',$,IFCMOMENTOFINERTIAMEASURE(0.00027),$);
#530= IFCPROPERTYSINGLEVALUE
('MomentOfInertiaZ',$,IFCMOMENTOFINERTIAMEASURE(0.00107),$);
#531= IFCPROPERTYSINGLEVALUE
('TorsionalSectionModulus',$,IFCSECTIONMODULUSMEASURE(0.622),$);
#532= IFCPROFILEPROPERTIES
('Pset_ProfileMechanical',$,(#527,#528,#529,#530,#531),#526);
#533= IFCEXTRUDEDAREASOLID (#526,#191,#13,5.3);
#534= IFCSHAPEREPRESENTATION (#202,'Body','SweptSolid',(#533));
#535= IFCPRESENTATIONLAYERASSIGNMENT ('Beam Profile','No description',(#534),$);
#536= IFCPRODUCTDEFINITIONSHAPE ($,$,(#534));
#537= IFCCARTESIANPOINT ((0.1,-0.05,2.8));
#538= IFCAXIS2PLACEMENT3D (#537,#9,#11);
#539= IFCLOCALPLACEMENT (#192,#538);
#540= IFCBEAM ('3zt8DTaE82lkeQ8FPj5gye',#5,'Beam 3',$,$,#539,#536,$,.BEAM.);
#541= IFCRELCONNECTSSTRUCTURALELEMENT
('3aDDAft390$A5guwnUv4xc',#5,$,$,(#371),#540);
#545= IFCCARTESIANPOINT ((0.1,5.55,2.8));
#546= IFCAXIS2PLACEMENT3D (#545,#9,#11);
#547= IFCLOCALPLACEMENT (#192,#546);
#548= IFCBEAM ('4zt8DTaE82lkeQ8FPj5gye',#5,'Beam 4',$,$,#547,#536,$,.BEAM.);
#549= IFCRELCONNECTSSTRUCTURALELEMENT
('4aDDAft390$A5guwnUv4xc',#5,$,$,(#381),#548);
#550= IFCRELCONTAINEDINSPATIALSTRUCTURE
('35kVXT9hb7C80NqRflZZF6',#5,$,$,(#540,#548),#196);

/* Laje - Secção e características */
#600= IFCRECTANGLEPROFILEDEF (.AREA.,$,$,5.3,5.4);
#601= IFCEXTRUDEDAREASOLID(#600,#191,#13,0.22);
#602= IFCSHAPEREPRESENTATION(#202,'Body','SweptSolid',(#601));
#603= IFCPRESENTATIONLAYERASSIGNMENT('Columns Profile','No description',(#602),$);
#604= IFCPRODUCTDEFINITIONSHAPE($,$,(#602));
#605= IFCPROPERTYSINGLEVALUE ('MassPerLength',$,IFCMASSPERLENGTHMEASURE(25),$);
```

```
#606= IFCPROPERTYSINGLEVALUE ('CrossSectionArea',$,IFCAREAMEASURE(1.21),$);
#607= IFCPROPERTYSINGLEVALUE
('MomentOfInertiaY',$,IFCMOMENTOFINERTIAMEASURE(3.05),$);
#608= IFCPROPERTYSINGLEVALUE
('MomentOfInertiaZ',$,IFCMOMENTOFINERTIAMEASURE(0.00488),$);
#609= IFCPROPERTYSINGLEVALUE
('TorsionalSectionModulus',$,IFCSECTIONMODULUSMEASURE(0.622),$);
#610= IFCPROFILEPROPERTIES
('Pset_ProfileMechanical',$,(#605,#606,#607,#608,#609),#600);
#611= IFCCARTESIANPOINT ((2.75,2.75,2.78));
#612= IFCAXIS2PLACEMENT3D (#611,#13,#9);
#613= IFCLOCALPLACEMENT (#192,#612);
#614= IFCSLAB ('1srmDN46zBVupUF9xps7Ev',#5,'Laje 1',$,$,#613,#604,$,.FLOOR.);
#615= IFCRELCONNECTSSTRUCTURALELEMENT
('9aDDAft390$A5gnwnUv1xc',#5,$,$,(#390),#614);
#616= IFCRELCONTAINEDINSPATIALSTRUCTURE
('45kVXT9hb7C80NqRflZZF6',#5,$,$,#614,#196);

/* ---------------------------------------------------------------------------- */


/* Material e propriedades */
/* Pilares - Propriedades do material */
#700= IFCMATERIAL ('C20/25',$,'Concrete');
#701= IFCPROPERTYSINGLEVALUE ('MassDensity',$,IFCMASSDENSITYMEASURE(2407.31),$);
#702= IFCMATERIALPROPERTIES ('Pset_MaterialCommon',$,(#701),#700);
#703= IFCPROPERTYSINGLEVALUE
('YoungModulus',$,IFCMODULUSOFELASTICITYMEASURE(23250000000),$);
#704= IFCPROPERTYSINGLEVALUE
('ShearModulus',$,IFCMODULUSOFELASTICITYMEASURE(9964000000),$);
#705= IFCMATERIALPROPERTIES ('Pset_MaterialMechanical',$,(#703,#704),#700);
#706= IFCPROPERTYSINGLEVALUE ('ConstructionMethod',$,'In-Situ',$);
#707= IFCPROPERTYSINGLEVALUE ('StructuralClass',$,'S1',$);
#708= IFCPROPERTYSINGLEVALUE ('StrengthClass',$,'C20/25',$);
#709= IFCPROPERTYSINGLEVALUE ('ExposureClass',$,'XC4',$);
#710= IFCPROPERTYSINGLEVALUE
('ReinforcementVolumeRatio',$,IFCMASSDENSITYMEASURE(280),$);
#711= IFCPROPERTYSINGLEVALUE ('ConstructionToleranceClass',$,'cm',$);
#712= IFCPROPERTYSINGLEVALUE ('ConcreteCover',$,IFCPOSITIVELENGHTMEASURE(0.025),$);
#713= IFCPROPERTYSINGLEVALUE ('ReinforcementStrengthClass',$,'S500',$);
#714= IFCPROPERTYSET
('75Mvg9UeH5Y9J2VB46x085',#5,'Pset_ConcreteElementGeneral',$,(#706,#707,#708,#709,#
710,#711,#712,#713));
#715= IFCRELDEFINESBYPROPERTIES
('1vnzm4QtjFRRfqpE2lddL5',#5,$,$,(#301,#311,#321,#331,#439,#444,#449,#454),#714);
#716= IFCMATERIALPROFILE ($,$,#700,#425,$,$);
#717= IFCMATERIALPROFILESET ($,$,(#716),$);
#718= IFCMATERIALPROFILESETUSAGE (#717,$,$);
#719= IFCRELASSOCIATESMATERIAL
('01kVXT9hb7C80NqRflZZF9',#5,$,$,(#301,#311,#321,#331,#439,#444,#449,#454),#718);

/* Vigas - Propriedades do material */
#750= IFCMATERIAL ('C20/25',$,'Concrete');
#751= IFCPROPERTYSINGLEVALUE ('MassDensity',$,IFCMASSDENSITYMEASURE(2407.31),$);
#752= IFCMATERIALPROPERTIES ('Pset_MaterialCommon',$,(#751),#750);
#753= IFCPROPERTYSINGLEVALUE
('YoungModulus',$,IFCMODULUSOFELASTICITYMEASURE(23250000000),$);
#754= IFCPROPERTYSINGLEVALUE
('ShearModulus',$,IFCMODULUSOFELASTICITYMEASURE(9964000000),$);
#755= IFCMATERIALPROPERTIES ('Pset_MaterialMechanical',$,(#753,#754),#750);
#756= IFCPROPERTYSINGLEVALUE ('ConstructionMethod',$,'In-Situ',$);
#757= IFCPROPERTYSINGLEVALUE ('StructuralClass',$,'S1',$);
#758= IFCPROPERTYSINGLEVALUE ('StrengthClass',$,'C20/25',$);
#759= IFCPROPERTYSINGLEVALUE ('ExposureClass',$,'XC4',$);
```

```
#760= IFCPROPERTYSINGLEVALUE
('ReinforcementVolumeRatio',$,IFCMASSDENSITYMEASURE(280),$);
#761= IFCPROPERTYSINGLEVALUE ('ConstructionToleranceClass',$,'cm',$);
#762= IFCPROPERTYSINGLEVALUE ('ConcreteCover',$,IFCPOSITIVELENGHTMEASURE(0.025),$);
#763= IFCPROPERTYSINGLEVALUE ('ReinforcementStrengthClass',$,'S500',$);
#764= IFCPROPERTYSET
('08Mvg9UeH5Y9J2VB46x085',#5,'Pset_ConcreteElementGeneral',$,(#756,#757,#758,#759,#
760,#761,#762,#763));
#765= IFCRELDEFINESBYPROPERTIES
('2vnzm4QtjFRRuupE2lddL3',#5,$,$,(#351,#361,#371,#381,#490,#495,#540,#548),#764);
#766= IFCMATERIALPROFILE ($,$,#750,#476,$,$);
#767= IFCMATERIALPROFILESET ($,$,(#766),$);
#768= IFCMATERIALPROFILESETUSAGE (#767,$,$);
#769= IFCRELASSOCIATESMATERIAL
('21kVXT9hb7C80NqRflZZF4',#5,$,$,(#351,#361,#371,#381,#490,#495,#540,#548),#768);


/* Lajes - Propriedades do material */
#806= IFCMATERIAL ('C20/25',$,'Concrete');
#807= IFCPROPERTYSINGLEVALUE ('MassDensity',$,IFCMASSDENSITYMEASURE(2407.31),$);
#808= IFCMATERIALPROPERTIES ('Pset_MaterialCommon',$,(#807),#806);
#809= IFCPROPERTYSINGLEVALUE
('YoungModulus',$,IFCMODULUSOFELASTICITYMEASURE(23250000000),$);
#810= IFCPROPERTYSINGLEVALUE
('ShearModulus',$,IFCMODULUSOFELASTICITYMEASURE(9964000000),$);
#811= IFCMATERIALPROPERTIES ('Pset_MaterialMechanical',$,(#809,#810),#806);
#812= IFCPROPERTYSINGLEVALUE ('ConstructionMethod',$,'In-Situ',$);
#813= IFCPROPERTYSINGLEVALUE ('StructuralClass',$,'S1',$);
#814= IFCPROPERTYSINGLEVALUE ('StrengthClass',$,'C20/25',$);
#815= IFCPROPERTYSINGLEVALUE ('ExposureClass',$,'XC4',$);
#816= IFCPROPERTYSINGLEVALUE
('ReinforcementVolumeRatio',$,IFCMASSDENSITYMEASURE(280),$);
#817= IFCPROPERTYSINGLEVALUE ('ConstructionToleranceClass',$,'cm',$);
#818= IFCPROPERTYSINGLEVALUE ('ConcreteCover',$,IFCPOSITIVELENGHTMEASURE(0.025),$);
#819= IFCPROPERTYSINGLEVALUE ('ReinforcementStrengthClass',$,'S500',$);
#820= IFCPROPERTYSET
('42Mvg9UeH5Y9J2VB46x085',#5,'Pset_ConcreteElementGeneral',$,(#812,#813,#814,#815,#
816,#817,#818,#819));
#821= IFCRELDEFINESBYPROPERTIES ('3vnzm4QtjFRRfqpE2lddL5',#5,$,$,(#390,#614),#820);
#822= IFCMATERIALPROFILE ($,$,#806,#600,$,$);
#823= IFCMATERIALPROFILESET ($,$,(#822),$);
#824= IFCMATERIALPROFILESETUSAGE (#823,$,$);
#825= IFCRELASSOCIATESMATERIAL ('31kVXT9hb7C80NqRflZZF6',#5,$,$,(#390,#614),#824);

/* Pilares - Identificação de Propriedades - Dimensões */
#850= IFCPROPERTYSINGLEVALUE ('a',$,IFCLENGTHMEASURE(0.2),$);
#851= IFCPROPERTYSINGLEVALUE ('b',$,IFCLENGTHMEASURE(0.3),$);
#852= IFCPROPERTYSET ('08Mvg9UeH5Y9J2VB46x085',#5,'Dimensions',$,(#850,#851));
#853= IFCRELDEFINESBYPROPERTIES
('2vnzm4QtjFRRuupE2lddL3',#5,$,$,(#439,#444,#449,#454),#852);

/* ---------------------------------------------------------------------------- */


/* Modelo de análise estrutural */
#875= IFCSTRUCTURALANALYSISMODEL ('0VYesmxUHFNez26MoJx5F3',#5,'Structural Analysis
1',$,$,.NOTDEFINED.,#191,(#900,#910,#920,#930),$,#192);
#876= IFCRELDECLARES ('1Frhk31Z12CR9mBbodz9XE',#5,'GROUP',$,#180,(#875));


/* Caso de carga 1 */
#900= IFCSTRUCTURALLOADCASE ('1fv4DZfY55exwX8QDy8dmw',#5,'Structural Load Case
1',$,$,.LOAD_CASE.,.NOTDEFINED.,.NOTDEFINED.,1.,$,(0.,0.,0.));

/* Pilar 1 - Carga linear */
```

```
#901= IFCSTRUCTURALLOADLINEARFORCE ('Nominal',$,$,-1.64,$,$,$);
#902= IFCSTRUCTURALLOADLINEARFORCE ('Nominal',$,$,-1.64,$,$,$);
#903= IFCSTRUCTURALLOADCONFIGURATION ($,(#901,#902),((0.),(3.)));
#904= IFCSTRUCTURALCURVEACTION ('1WSwGyLsrFNA9TLOq_ifyd',#5,'Structural Curve
Action 1',$,$,$,$,#903,.GLOBAL_COORDS.,.F.,$,.LINEAR.);
#905= IFCRELCONNECTSSTRUCTURALACTIVITY ('0XvroPpOb4FPsGBZQ$pgtA',#5,$,$,#301,#904);
#906= IFCRELASSIGNSTOGROUP ('1OygXKIkL35eDtUalQjese',#5,$,$,(#904),.PRODUCT.,#900);


/* Caso de carga 2 */
#910= IFCSTRUCTURALLOADCASE ('2fv4DZfY55exwX8QDy8dmw',#5,'Structural Load Case
2',$,$,.LOAD_CASE.,.NOTDEFINED.,.NOTDEFINED.,1.,$,(0.,0.,0.));

/* Pilar 1 - Carga pontual */
#911= IFCSTRUCTURALLOADSINGLEFORCE ('Força 1',$,$,-2.54,$,$,$);
#912= IFCSTRUCTURALLOADCONFIGURATION ($,(#911),((3.)));
#913= IFCSTRUCTURALPOINTACTION ('2WSwGyLsrFNA9TLOq_ifyd',#5,'Structural Point
Action 1',$,$,$,$,#912,.GLOBAL_COORDS.,.F.);
#914= IFCRELCONNECTSSTRUCTURALACTIVITY ('1XvroPpOb4FPsGBZQ$pgtA',#5,$,$,#321,#913);
#915= IFCRELASSIGNSTOGROUP ('2OygXKIkL35eDtUalQjese',#5,$,$,(#913),.PRODUCT.,#910);


/* Caso de carga 3 */
#920= IFCSTRUCTURALLOADCASE ('3fv4DZfY55exwX8QDy8dmw',#5,'Structural Load Case
3',$,$,.LOAD_CASE.,.NOTDEFINED.,.NOTDEFINED.,1.,$,(0.,0.,0.));

/* Viga 1 - Carga trapezoidal */
#921= IFCSTRUCTURALLOADLINEARFORCE ('Nominal',5,$,$,$,$,$);
#922= IFCSTRUCTURALLOADLINEARFORCE ('Nominal',2,$,$,$,$,$);
#923= IFCSTRUCTURALLOADCONFIGURATION ($,(#921,#922),((0.),(5.5)));
#924= IFCSTRUCTURALCURVEACTION ('3WSwGyLsrFNA9TLOq_ifyd',#5,'Structural Curve
Action 2',$,$,$,$,#923,.GLOBAL_COORDS.,.F.,$,.LINEAR.);
#925= IFCRELCONNECTSSTRUCTURALACTIVITY ('6XvroPpOb4FPsGBZQ$pgtA',#5,$,$,#351,#924);
#926= IFCRELASSIGNSTOGROUP ('3OygXKIkL35eDtUalQjese',#5,$,$,(#924),.PRODUCT.,#920);


/* Caso de carga 4 */
#930= IFCSTRUCTURALLOADCASE ('4fv4DZfY55exwX8QDy8dmw',#5,'Structural Load Case
4',$,$,.LOAD_CASE.,.NOTDEFINED.,.NOTDEFINED.,1.,$,(0.,0.,0.));

/* Laje 1 - Carga uniformemente distribuida */
#931= IFCSTRUCTURALLOADPLANARFORCE ('Nominal',$,$,-4);
#932= IFCSTRUCTURALLOADCONFIGURATION ($,(#931),((0.)));
#933= IFCSTRUCTURALPLANARACTION ('4WSwGyLsrFNA9TLOq_ifyd',#5,'Structural Planar
Action 1',$,$,$,$,#932,.GLOBAL_COORDS.,.F.,$,.CONST.);
#934= IFCRELCONNECTSSTRUCTURALACTIVITY ('0XvroPpOb4FPsGBZQ$pgtA',#5,$,$,#390,#933);
#935= IFCRELASSIGNSTOGROUP ('4OygXKIkL35eDtUalQjese',#5,$,$,(#933),.PRODUCT.,#930);


ENDSEC;

END-ISO-10303-21;
```