IFC – Desafios de exportação e resolução com Python IfcOpenShell

https://doi.org/10.21814/uminho.ed.164.12

Hugo Silva[1], **Luís Ribeirinho**[1], **Sofia Henriques**[1]

[1] *TPF – Consultores de Engenharia e Arquitetura, S.A., Lisboa*

Resumo

A TPF – Consultores de Engenharia e Arquitetura, S.A. tem desenvolvido diversos projetos em openBIM, nos quais se deparou com vários desafios de exportação de *Revit* para IFC.

Por vezes, as ferramentas e opções de exportação nativas dos programas de modelação falham, impondo-se assim a necessidade de se encontrarem soluções que deem cumprimento aos requisitos de transmissão de informação no formato IFC.

Num projeto específico que envolveu vários hospitais semelhantes, foi necessário federar diversos edifícios num só ficheiro IFC e articular a posição de cada um dos edifícios, de acordo com o hospital em questão. Para superar este desafio, que não pode ser resolvido a partir do modelo nativo, recorreu-se a *scripts* em linguagem *Python*, utilizando a biblioteca IfcOpenShell.

No *Revit*, é possível configurar a classificação das entidades IFC que se atribui a cada objeto através dos parâmetros IfcExportAs e IfcExportType. Contudo, existem limitações reconhecidas pela Autodesk para determinadas categorias. Este problema foi também solucionado com um *script* em linguagem *Python*, corrigindo o IFC exportado do *Revit*, e adicionando a informação que não foi transmitida para o IFC.

1. Introdução

Num contexto em que a comunicação eficiente dos dados contidos nos modelos BIM entre diversas plataformas de modelação e ferramentas BIM é cada vez mais crucial na indústria AECO (Arquitetura, Engenharia, Construção e Operação), o formato IFC [2] sobressai como um elemento vital para possibilitar essa interoperabilidade. Nesse sentido, a correta estruturação e apresentação da informação relevante tornam-se imperativas, seguindo regras cada vez mais rigorosas para atender aos requisitos de comunicação e interoperabilidade de forma eficaz e eficiente.

Embora os principais programas de modelação tenham progredido consideravelmente na robustez das operações de exportação e importação de IFCs nos respetivos modelos nativos, há desafios frequentes que persistem, exigindo soluções para além das capacidades destas plataformas tecnológicas. Embora existam ferramentas disponíveis que permitem a edição de ficheiros IFC, como o *BIMvision* e o *usBIM*, é importante reconhecer que essas soluções podem apresentar limitações. Muitas vezes, a necessidade de intervenção manual extensa pode tornar os processos morosos e suscetíveis a erros, além de por vezes não oferecerem recursos avançados para lidar com a complexidade dos dados contidos nos modelos IFC.

O formato IFC, baseado na estrutura física de ficheiro STEP "Standard for the Exchange of Product Data" de acordo com a norma ISO 10303-21 [3], apresenta-se também como um ficheiro de texto ASCII editável. Compreendendo as diversas ontologias que agrupam os objetos, classes, atributos e seus relacionamentos, definidas por uma estrutura de dados preconizadas pela *buildingSMART* (*schema* do IFC), é possível contornar, como último recurso, algumas das limitações na exportação IFC que por vezes surgem nos programas de modelação, através da edição direta dos ficheiros .ifc, enquanto ficheiros de texto.

Nesta abordagem, é possível automatizar a edição dos ficheiros IFC recorrendo a *scripts* em linguagem *Python* [1], que, utilizando a biblioteca IfcOpenShell, permitem simplificar muitos passos envolvidos nesse processo. Este princípio oferece uma abordagem flexível para lidar com desafios específicos e aprimorar a interoperabilidade entre plataformas BIM.

2. Caso de estudo

A TPF – Consultores de Engenharia e Arquitetura, S.A., desempenhou o papel de Entidade Fornecedora Líder num projeto multidisciplinar em openBIM de quatro hospitais, desenvolvidos em *Revit* 2022. Um dos entregáveis cruciais para este projeto era o modelo em formato IFC, com o *schema* IFC4 ADD2 TC1. Destes quatro hospitais, três partilham a mesma solução conceptual, variando apenas a posição dos vários edifícios que constituem cada unidade hospitalar, com a particularidade de alguns edifícios se repetirem.

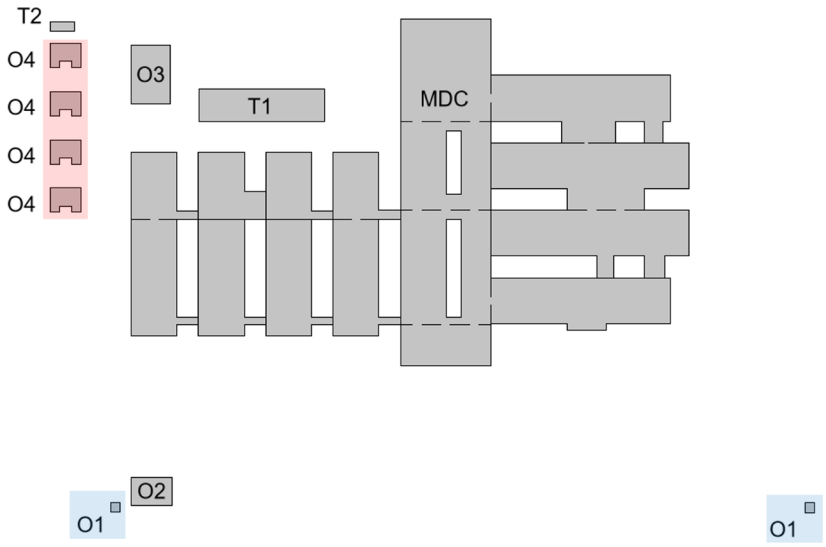

Figura 1
Diagrama dos edifícios que compõem cada hospital.

Conforme ilustrado na Figura 1, cada hospital compreende 11 edifícios (edifício principal + edifícios satélites). Destaca-se que o edifício satélite O1 se repete 2 vezes, enquanto o edifício O4 ocorre em quatro instâncias.

Considerando que os marcos de entrega do projeto e o ritmo de construção de cada hospital são discrepantes, optou-se por uma estratégia de modelação em que cada tipo de edifício foi modelado separadamente. Esses modelos foram posteriormente ligados a modelos agregadores específicos de cada hospital, considerando as suas respetivas localizações. Essa abordagem permitiu que à medida que os sete modelos (O1 a O4, T1, T2 e MDC) evoluíam, as atualizações eram automaticamente refletidas nos três hospitais, tornando o processo substancialmente mais eficiente, economizando tempo e esforço para todas as equipas envolvidas.

No entanto, numa fase avançada do projeto, a Entidade Requerente recusou aceitar 11 modelos IFC por especialidade para cada hospital, ou um ficheiro de cache de um visualizador IFC que agregasse os diversos modelos. Dado que o *Revit 2022* não permite a exportação de *links* para um único ficheiro IFC, esta situação tornou-se um desafio que necessitava de uma solução.

Para contornar este obstáculo, foi desenvolvido um automatismo em *Python* para manipular os IFCs exportados do *Revit* e combiná-los num único ficheiro IFC.

3. Manipulação de IFCs no caso de estudo

Antes de se iniciar o desenvolvimento do *script*, foi testado o processo de junção de IFCs num único ficheiro através do *BIMvision* com o módulo IFC Merge. Para utilizar este sotfware foi necessário exportar para IFC todos os modelos em triplicado para cada uma das localizações, e repetir o processo para os modelos satélite que se repetiam no mesmo hospital.

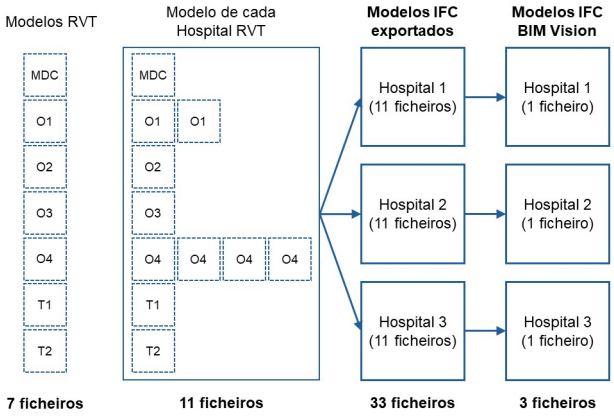

Figura 2
Modelos IFC Exportados para utilização no BIMvision.

Como mostra a Figura 2, caso este método fosse aplicado, o processo de exportação de IFCs implicaria gerar 33 ficheiros IFC por disciplina a cada entrega. Como na altura, o BIMvision não tinha a opção de gerar novos GUIDs, os elementos dos edifícios satélite que se repetiam ficavam com o mesmo GUID, o que gerava um conflito de elementos no momento da sua junção. Seria possível a alteração dos GUIDs, como se acabou por fazer com o *script* (que se explica abaixo), mas a exportação dos 33 IFCs a cada entrega seria demorado e sujeito a erros na escolha correta de cada localização para cada modelo *Revit* no momento da exportação.

A seguir foi estudado o processo através da edição manual dos vários IFCs em modo de ficheiro de texto, a fim de determinar o melhor fluxo a adotar para a automatização. Determinou-se que copiando a informação da "data section" de um segundo ficheiro IFC, com alteração prévia das numerações dos STEP ids de modo que não houvesse repetições com os ids do primeiro modelo IFC para onde esta informação é copiada, estes dois modelos ficam unidos num único ficheiro.

Seguindo este princípio, adicionaram-se mais modelos ao primeiro modelo. Contudo, era importante ajustar a estrutura do novo IFC. Este ajuste visava a eliminação dos IfcProject e os IfcSite em excesso e associar os vários IfcBuilding ao mesmo IfcSite, assegurando uma estrutura hierárquica correta do IFC.

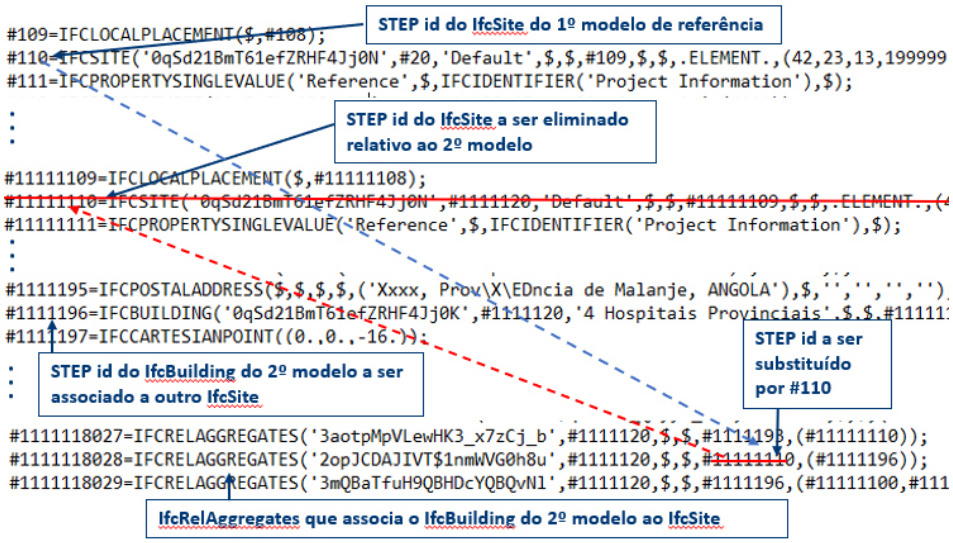

Figura 3
Exemplo de associação IfcBuilding a outro IfcSite.

Para realizar essa associação, foi necessário identificar o STEP id do IfcBuilding de cada modelo adicional e do seu IfcRelAggregates, que o relaciona ao IfcSite. De seguida, substituiu-se o número do STEP id referente ao IfcSite do primeiro modelo de referência, conforme ilustrado na Figura 3.

Cada instância da classe IfcBuildingStorey possui um IfcObjectPlacement definido com coordenadas relativas ao IfcBuilding, enquanto o posicionamento de cada objeto é determinado pelo IfcObjectPlacement associado ao IfcBuildingStorey correspondente. Portanto, a localização de todos os elementos está intrinsecamente ligada à posição global do IfcBuilding.

Assim, para ajustar a posição dos elementos de cada edifício, adicionaram-se duas linhas/instâncias ao ficheiro, IfcCartesianPoint e IfcAxis2Placement3D, onde se definiram novas coordenadas e associou-se o seu STEP id ao respetivo IfcLocalPlacement que posiciona o IfcBuilding.

No âmbito deste caso de estudo, o objetivo era exportar apenas os sete ficheiros IFC e, a partir destes, criar automaticamente um IFC para cada um dos três hospitais,composto pelos 11 edifícios,através do automatismo proposto – ver Figura 4.

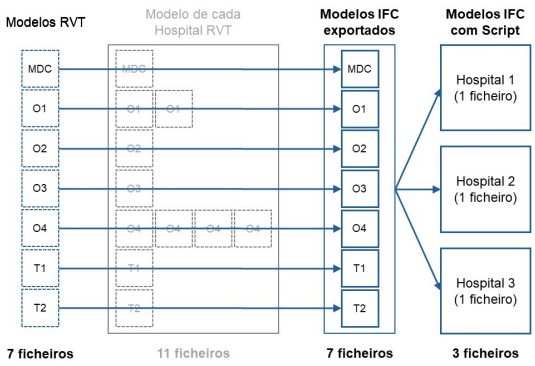

Figura 4
Modelos IFC Exportados para utilização pelo Script.

Em cada iteração, o ficheiro IFC editado foi aberto em cinco visualizadores IFC distintos de referência (BIMCollab ZOOM, Solibri, BIMvision, usBIM e XbimXplorer) para verificar e garantir que poderia ser utilizado com sucesso de uma forma abrangente.

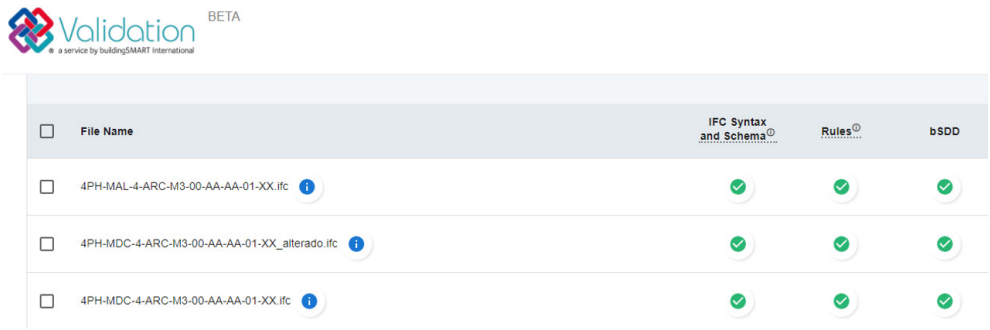

Figura 5
Relatórios do serviço de validação de IFCs disponibilizado pela buildingSMART.

Tendo o processo estabilizado, foi também feita uma verificação final utilizando o serviço da buildingSMART de validação de modelos IFC. Na Figura 5, encontram-se alguns relatórios retirados deste serviço da buildingSMART.

3.1. Automatismo: Alterar a informação nos modelos

A fase inicial do processo automatizado envolve a leitura do nome dos 7 ficheiros IFC exportados, os quais estão localizados numa pasta específica para este propósito. A partir desses ficheiros exportados, o *script* cria ficheiros IFC temporários, duplicando os ficheiros para os edifícios que se repetem e adicionando um prefixo a cada ficheiro para distinguir todos 11 edifícios.

Durante a criação destes ficheiros IFC temporários, são efetuadas alterações nas propriedades de cada modelo individual, incluindo o nome do IfcBuilding e a sua posição relativa. Para realizar essas modificações, foram empregues dicionários que associam a codificação destes ficheiros às coordenadas específicas onde cada edifício deve ser posicionado, dependendo do hospital em questão.

A utilização do IfcOpenShell simplificou significativamente o *script* no reposicionamento dos edifícios. Uma única linha de código realizou o que seria necessário implementar em várias linhas de código utilizando a abordagem clássica de edição das linhas de texto do ficheiro, no formato .txt.

Numa nota importante, alguns dos modelos temporários eram cópias da mesma exportação (edifícios O1 e O4), resultando em objetos com o mesmo GUID. Antes de se juntar esses modelos num único ficheiro, tornou-se necessário gerar novos GUIDs, garantindo assim que não existiriam elementos com GUIDs duplicados e salvaguardando a utilização bem-sucedida do novo ficheiro federado.

3.2. Automatismo: Unir modelos

Com a informação devidamente organizada nos modelos temporários, o *script* procede à união destes modelos, gerando um novo ficheiro de raiz que incorpora cada instância de cada modelo temporário. Neste processo, a ferramenta IfcOpenShell é novamente empregue, não só para unir os edifícios/modelos num único ficheiro, mas também para remover as entidades IfcProject e IfcSite dos modelos IFC agrupados, com a exceção do primeiro ficheiro da lista. Este procedimento associa todos os IfcBuilding ao mesmo IfcSite e IfcProject.Ao finalizar, o *script* elimina os IFCs temporários que serviram de base à criação do modelo IFC federado num único ficheiro.

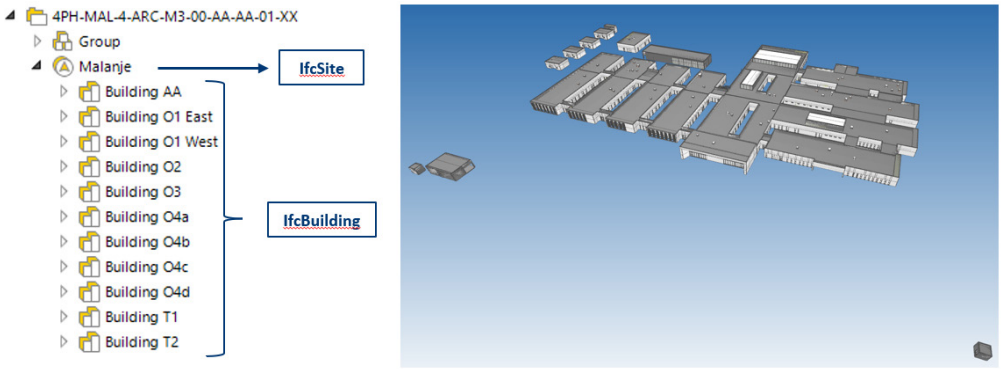

Figura 6
Modelo IFC resultante do *script*, visualizado no BIMCollab ZOOM.

Conforme a Figura 6 ilustra, a estrutura do modelo federado IFC segue uma hierarquia de IfcProject > IfcSite > IfcBuilding 1 + IfcBuilding 2 +...+ IfcBuilding *n*, ou seja, um só IfcSite que contém os diferentes IfcBuilding, conforme previsto no *schema* da buildingSMART. Apesar de comprovadamente funcionar nos vários visualizadores IFC, a Entidade Requerente, que opera em *Archicad 15* e é responsável por preparar os modelos dos equipamentos médicos, reportou a impossibilidade de importar todos os edifícios para este programa. Esta versão do *Archicad,* permite a importação de apenas um IfcBuilding por IFC, limitando a utilização dos modelos com esta estrutura, especialmente os IFCs de Arquitetura e Estrutura, que precisavam ser importados no *Archicad*.

Face a essa limitação, foi adicionada uma variante ao *script*. A partir do novo modelo IFC federado, são reunidos todos os relacionamentos IfcRelContainedInSpatialStructure, realocando os elementos aos IfcBuildingStorey originários do primeiro modelo temporário.

Para isso, foi crucial o cumprimento do BEP, especificamente na definição dos IfcBuildingStorey, onde todos os IfcBuilding seguem a mesma estrutura de IfcBuildingStorey, nomeadamente a sua nomenclatura.

Considerando a intenção de alocar cada objeto a um outro IfcBuildingStorey com o mesmo nome onde está atualmente alocado, o *script* correlaciona o nome do RelatingStructure de cada IfcRelContainedInSpatialStructure através de ciclos contados

(*loops*). Nessa correlação, se for verificado que o nome do seu RelatingStructure é igual ao nome de um dos IfcBuildingStoreys de referência, a relação espacial é atribuída ao IfcBuildingStorey correspondente.

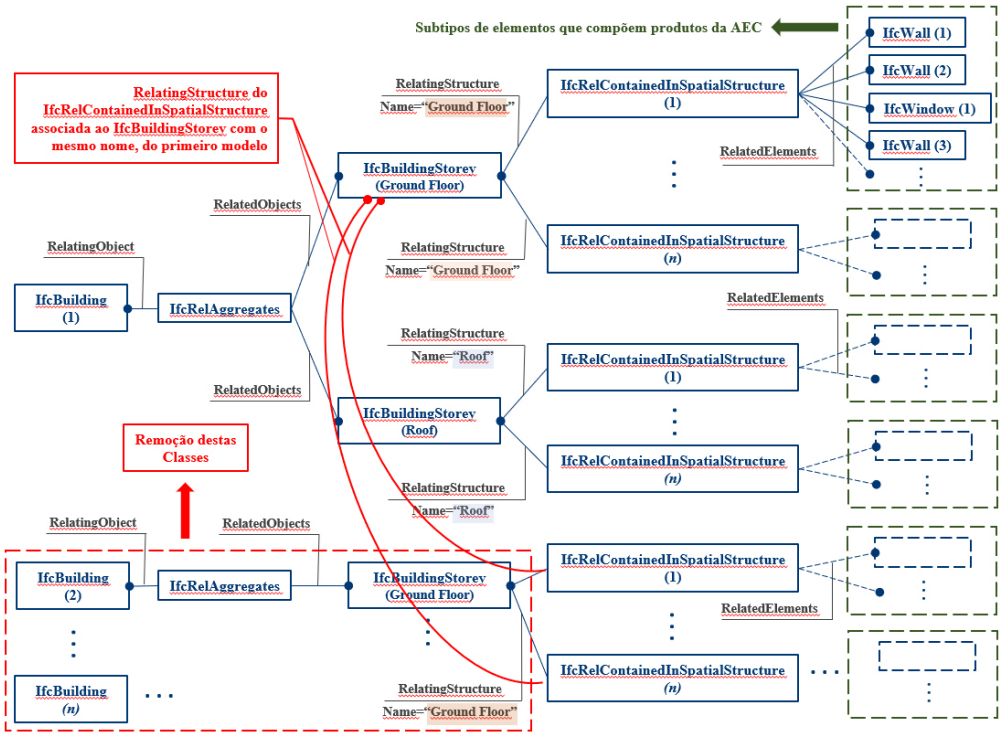

Figura 7
Esquema de realocação de objetos a outro IfcBuildingStorey.

Na Figura 7 pode ser encontrado um esquema representativo da realocação de objetos a outro IfcBuildingStorey.

Após a alocação de todos os objetos a um único IfcBuilding, nos respetivos IfcBuildingStorey, o *script* remove os outros IfcBuilding e respetivos IfcBuildingStorey (que ficaram sem objetos alocados).

Para melhor ilustração, a Figura 8 apresenta a estrutura IFC originada por esta esta variante do *script*, em comparação à estrutura IFC criada pelo *script* original.

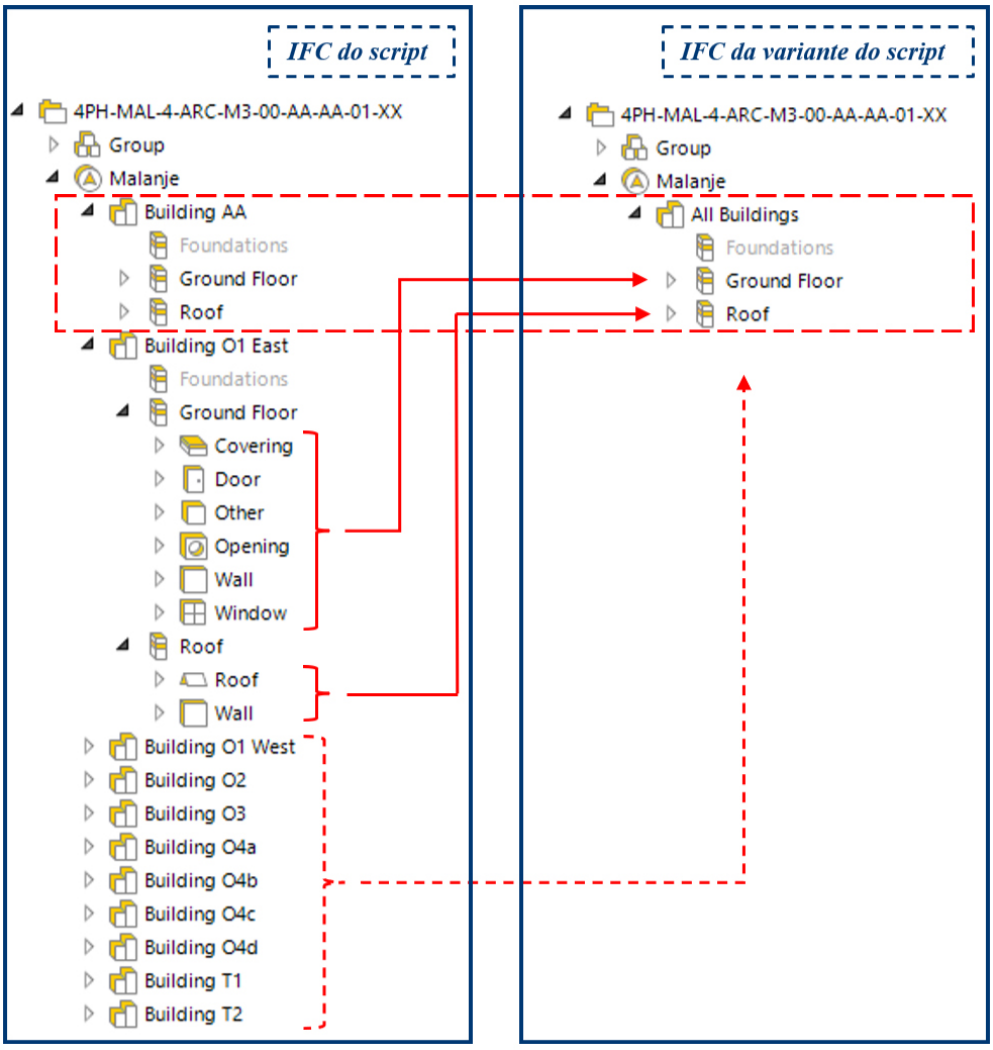

Figura 8
Comparação das duas estruturas IFC resultantes do *Script*, visualizado no BIMCollab ZOOM.

4. Outras aplicações de manipulação de IFCs

Além das diversas manipulações aos IFCs descritas no capítulo anterior, foram também desenvolvidos outros *scripts* para lidar com situações pontuais.

4.1. Classificação IFC

Nos modelos *Revit* da disciplina de Arquitetura, os capeamentos de parapeitos são frequentemente modelados na categoria *Revit "Wall Sweeps"*. No entanto, a *Autodesk* reconhece uma lacuna não resolvida no *Revit* quanto à exportação destes elementos para a classificação IFC adequada. A opção de se utilizar o parâmetro IfcExportAs para atribuir a classificação IFC não funciona para esta categoria de objeto, resultando com que este seja gerado como IfcBuildingElementProxy, ou seja, sem classificação IFC.

Considerando que a classificação IFC desejada para esses objetos é IfcCovering, que partilha das mesmas instâncias hierárquicas do IfcBuildingElementProxy, a TPF

Consultores desenvolveu um *script* que edita o ficheiro em formato .txt, aplicando essa classificação. Este *script* substitui o texto "IFCBUILDINGELEMENTPROXY" por "IFCCOVERING", nas linhas de texto que contêm esta classificação (IfcBuildingElementProxy), juntamente com a palavra "Capeamento".

Além disso, o *script* utiliza IfcOpenShell para criar o Pset_CoveringCommon previsto na buildingSMART, e adiciona a informação em falta a este Pset que não foi transferida do modelo *Revit* para o IFC. Essa abordagem garante a correção da classificação e a inclusão da informação necessária, mantendo a qualidade dos modelos IFC resultantes.

4.2. Alteração de cor de elementos IFC

Num caso muito específico, o Cliente solicitou a alteração das cores de paredes e tetos nos modelos IFC de arquitetura, visando uma apresentação importante que se iria realizar nesse mesmo dia, durante a visita de um ministro. Dada à limitação do tempo disponível para atender a esse pedido, que era insuficiente para abrir todos os modelos *Revit*, alterar as definições de materiais (cores) e fazer novas exportações IFC, optou-se por editar diretamente os IFCs que já tinham sido entregues anteriormente.

Para responder a esse desafio, foram desenvolvidas algumas linhas de código em *Python,* capazes de editar cada ficheiro IFC, enquanto formato .txt, procurando nas classes IfcColourRGB a cor que se pretendia alterar e efetuar a alteração para a cor desejada. De notar que o IFC utiliza o sistema de cores RGB, mas essas cores são definidas entre 0 e 1, em contraste com o intervalo de números inteiros de 0 a 255 utilizado para definir cores.

5. Futuros desenvolvimentos

Apesar das novas possibilidades de exportação de um modelo *Revit* na versão 2024 com *links* para um único ficheiro IFC, esta funcionalidade ainda apresenta problemas e erros no momento da exportação. No *BIMvision* também já existe a possibilidade de gerar novos GUIDs para os elementos combinados num único ficheiro IFC, mas para situações como o do caso de estudo apresentado continua a ser necessário a exportação dos 33 ficheiros como descrito no Capítulo 3. Por isso a TPF continua a explorar e desenvolver ferramentas que auxiliem este processo, e que permitam maior controlo e qualidade dos IFCs gerados.

Atualmente, está em andamento a implementação de um fluxo com JSON, com o propósito de armazenar os GUIDs dos modelos exportados correlacionados aos novos GUIDs gerados pelo *script*. Este processo visa preservar o novo GUID de cada objeto, mantendo uma base de dados onde novos GUIDs são atribuídos apenas a objetos que ainda não passaram pelo *script*. Embora a variação de GUIDs não seja uma condicionante para a Entidade Requerente do caso de estudo apresentado neste artigo, esta melhoria tem como objetivo entregar os modelos IFC sem que haja variação

nos mesmos objetos partilhados em momentos diferentes. A manutenção do mesmo GUID em cada objeto revela-se particularmente relevante para assegurar uma rastreabilidade eficaz de colisões/conflitos comunicados através do formato BCF.

6. Conclusões

O desenvolvimento de *scripts Python* para a manipulação eficiente de modelos IFC revelaram-se cruciais para enfrentar desafios específicos no contexto de entregas de projeto de engenharia e arquitetura em openBIM. A utilização da biblioteca IfcOpenShell demonstrou ser uma ferramenta valiosa, simplificando tarefas complexas e possibilitando ajustes precisos nos modelos IFC.

A automatização do processo de união de modelos IFC, proporcionou uma assimilação mais eficaz entre diferentes hospitais num projeto multidisciplinar de grande escala em openBIM. A capacidade de lidar com limitações específicas de determinadas plataformas, como a importação no *Archicad*, evidenciou a flexibilidade e adaptabilidade dos *scripts* desenvolvidos.

Além disso, a criação de *scripts* para classificação IFC e a alteração dinâmica de cores em modelos IFC num tempo muito curto demonstrou a versatilidade e rapidez que a automação proporciona, atendendo a requisitos específicos do Cliente.

Em síntese, os *scripts* desenvolvidos não apenas otimizaram processos operacionais, mas também demonstraram ser uma resposta ágil e eficaz para cenários específicos em que os programas de modelação não conseguem dar resposta. Ao fazer isso, proporcionam uma maior flexibilidade e robustez aos projetos de engenharia e arquitetura, especialmente quando envolvem a interoperabilidade de modelos BIM através do formato IFC.

Referências

[1] João Pavão Martins, *Programação em PYTHON: Introdução à programação utilizando múltiplos paradigmas*. Instituto Superior Técnico, 2020.

[2] ISO/TC59/SC13, "ISO 16739-1:2018 Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries", International Organization for Standardization (ISO),2018.

[3] ISO/TC 184/SC 4, "ISO 10303-21:2016 Industrial automation systems and integration — Product data representation and exchange — Part 21: Implementation methods: Clear text encoding of the exchange structure", International Organization for Standardization (ISO), 2016.