XXI ENTAC – Encontro Nacional de Tecnologia do Ambiente Construído
14 a 16 de outubro de 2026 | Goiânia-Brasil
DOI: https://doi.org/10.46421/entac.v21i1.8404

Geração Automatizada de Modelo de Pórtico Equivalente para Alvenaria Estrutural a partir de Arquivos IFC

Automated Generation of Equivalent Frame Models for Structural Masonry from IFC Files

Bruno de Oliveira Furlam
Universidade de São Paulo | São Paulo | Brasil | buno.furlam@usp.br
Fabiano Rogerio Corrêa
Universidade de São Paulo | São Paulo | Brasil | fabiano.correa@usp.br

Resumo

A integração entre modelos arquitetônicos e softwares estruturais enfrenta gargalos de interoperabilidade, exigindo o retrabalho manual de modelos analíticos. Sob a ótica da *Design Science Research* (DSR), este artigo apresenta um artefato tecnológico em Python para a conversão automatizada de modelos físicos de alvenaria estrutural em modelos de Pórtico Equivalente. O método utiliza a biblioteca IfcOpenShell para extrair diretrizes nativas e relacionamentos topológicos de arquivos IFC. O algoritmo deduz eixos, lintéis e nós, gerando o domínio analítico sem processamento complexo de geometria volumétrica. A aplicação em um estudo de caso de 3.575 m² atestou a geração de uma malha nodal coerente e a correta atribuição de seções transversais, com validação no software BIMCollab Zoom e tempo de processamento de 20,52 segundos. Conclui-se que o método viabiliza a interoperabilidade fluida entre disciplinas, mitigando a dependência de conversores proprietários e consolidando o uso do *openBIM* no projeto estrutural.

Palavras-chave: Alvenaria estrutural. OpenBIM. Interoperabilidade. Pórtico equivalente. Python.

Abstract

The integration between architectural models and structural analysis software faces interoperability bottlenecks, requiring manual rework of analytical models. Grounded in Design Science Research (DSR), this paper presents a Python-based technological artifact for the automated conversion of structural masonry physical models into Equivalent Frame models. The method uses the IfcOpenShell library to extract native guidelines and topological relationships from IFC files. The algorithm deduces axes, lintels, and nodes, generating the analytical domain without complex volumetric geometry processing. Application in a 3,575 m² case study confirmed a coherent nodal mesh and correct cross-section assignment, validated via BIMCollab Zoom with a processing time of 20.52 seconds. It is concluded that the method enables fluid interoperability between disciplines, mitigating reliance on proprietary converters and consolidating openBIM for structural design.

Keywords: Structural masonry. OpenBIM. Interoperability. Equivalent frame. Python.

INTRODUÇÃO

A alvenaria estrutural consolidou-se no cenário da construção civil brasileira como um sistema racionalizado e de alta produtividade, sendo amplamente adotada em empreendimentos habitacionais devido à sua eficiência econômica [1]. Paralelamente,

Como citar:
FURLAM, B. O; CORRÊA, F. R. Geração Automatizada de Modelo de Pórtico Equivalente para Alvenaria Estrutural a partir de Arquivos IFC. In: ENCONTRO NACIONAL DE TECNOLOGIA DO AMBIENTE CONSTRUÍDO, 21, 2026, Goiânia. **Anais**... Goiânia: ANTAC, 2026.

a disseminação da Modelagem da Informação da Construção (BIM) transformou a gestão de projetos, permitindo uma compatibilização espacial rigorosa entre as disciplinas de arquitetura e sistemas prediais [2]. Contudo, a integração efetiva entre os modelos IFC arquitetônicos e o projeto estrutural permanece como um desafio técnico, restrita à detecção de interferências geométricas em detrimento de uma interoperabilidade funcional.

As abordagens documentadas na literatura para mitigar o gargalo da transição entre o modelo físico (3D) e o modelo analítico são diversas. Pesquisadores têm explorado arquivos intermediários em formatos XML ou JSON para garantir a universalidade da troca de dados [3][4][5][6], além de algoritmos fundamentados na extração de topologias de Representação de Fronteira (BRep) para assegurar a integridade matemática da malha [7]. Mais recentemente, a literatura aponta o emprego de tecnologias emergentes, como motores de inferência ontológica [8], Web Semântica e algoritmos de Aprendizado de Máquina para inferir parâmetros estruturais a partir de contextos arquitetônicos [9].

Apesar desses avanços globais focados em algoritmos complexos, a prática comercial brasileira de projetos de alvenaria estrutural consolidou-se em torno de métodos mais simplificados e de menor custo computacional. No domínio do cálculo, o Método dos Elementos Finitos (MEF) destaca-se pelo rigor analítico na discretização de painéis laminares, mas é limitado pelo elevado tempo de processamento das malhas [10]. Assim, o Modelo de Pórtico Equivalente (MPE) para a obtenção de esforços horizontais, associado ao Modelo de Grupos de Paredes (MGP) para esforços verticais, tornaram-se a tipologia central em softwares hegemônicos no Brasil, como Eberick e TQS [11][12]. O MPE idealiza as paredes como pórticos tridimensionais de barras e nós, apresentando resultados de cálculo equivalentes aos do MEF com um custo-benefício superior em termos de modelagem e hardware [10].

Atualmente, nos softwares comerciais citados, o IFC atua apenas como referência para o lançamento manual da estrutura. Já em outros programas que suportam a importação do arquivo, como Robot e Rfem, exige-se que o modelo seja previamente convertido para o domínio de análise estrutural (*IfcStructuralAnalysisDomain*) [13][14].

Para preencher essa lacuna, esta pesquisa adota o método *Design Science Research* para desenvolver e avaliar um artefato tecnológico fundamentado na biblioteca *open-source* IfcOpenShell [15]. O objetivo é automatizar a geração do modelo estrutural a partir do arquivo IFC arquitetônico, transformando dados do *IfcArchitecturalDomain* e *IfcStructuralElementsDomain* para o *IfcStructuralAnalysisDomain*. Com esta abordagem, procura-se consolidar um método eficiente para a conversão direta entre os domínios geométrico e analítico, viabilizando a interoperabilidade funcional e reduzindo a dependência de processos manuais no projeto de alvenaria estrutural.

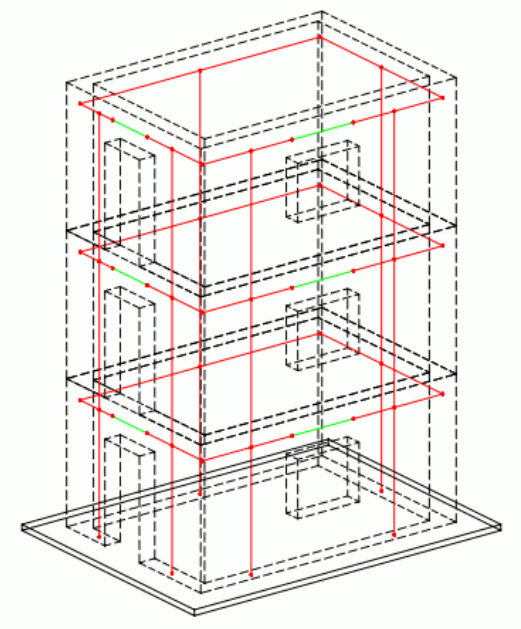
DOI: https://doi.org/10.46421/entac.v21i1.8404

O MODELO DO PÓRTICO EQUIVALENTE (MPE)

O MPE fundamenta-se na discretização da estrutura de alvenaria, substituindo os painéis contínuos por um sistema reticulado composto por barras. Nessa modelagem, a geometria bidimensional das paredes é simplificada para elementos unidimensionais, onde cada trecho de parede é idealizado como um pilar vertical posicionado exatamente em seu eixo baricêntrico. Simultaneamente, os lintéis — que compreendem o conjunto de vergas e peitoris transpondo os vãos — são representados como barras horizontais que conectam as paredes [16]. Essa transformação permite que a estabilidade global e a distribuição de esforços horizontais, como a ação do vento, sejam avaliadas por meio de análise matricial.

Para refletir o comportamento físico real, corrige-se a discrepância entre o nó teórico da barra e a largura real da parede. Como os painéis de alvenaria possuem dimensões transversais elevadas, o lintel não se conecta ao pilar em um ponto central isolado, mas sim na face lateral da abertura, distante do centroide. Para solucionar essa questão, o modelo incorpora barras horizontais: segmentos de rigidez axial infinita que se estendem do eixo baricêntrico da parede até a face do vão [16]. O uso desses braços é indispensável para transmitir as rotações e deslocamentos entre os elementos, garantindo que o modelo considere o enrijecimento provocado pela massa da alvenaria nas zonas de encontro e evite uma flexibilidade irreal no sistema.

Figura 1: Exemplificação do Modelo de Pórtico Equivalente para alvenaria.



Fonte: [16].

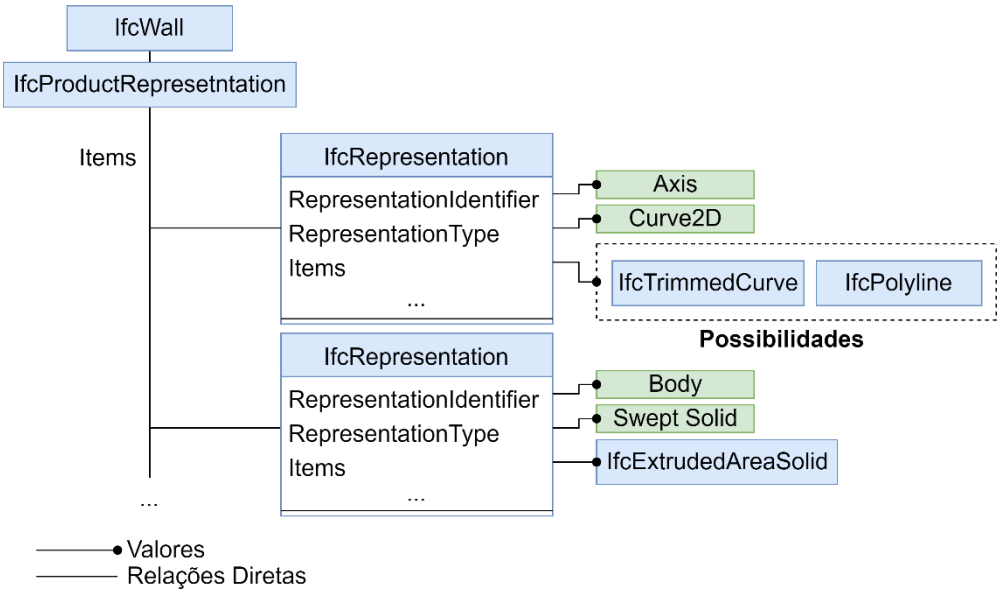
ESQUEMA IFC E REQUISITOS DE MODELAGEM

O algoritmo proposto delimita-se ao uso das versões IFC 4 e IFC 4.3. Como o esquema IFC integral é vasto, a buildingSMART estabelece subconjuntos denominados *Model View Definitions* (MVDs) [17]. O foco do fluxo reside na interpretação funcional dessas

entidades, priorizando os dados semânticos estruturais em detrimento de interpretações sólido-espaciais complexas, apoiando-se em um subconjunto do MVD *Reference View* (RV) [18][19].

A *Reference View* define diversas formas de representação geométrica, porém, para a extração da topologia estrutural, o conceito principal é o de eixo. Elementos lineares, como paredes, possuem uma representação volumétrica (*Body*) e uma representação bidimensional simplificada em planta (*Axis*). Essa representação em eixo é o ponto de partida do algoritmo, pois localiza o conjunto de camadas (*IfcMaterialLayerSet*), responsável por definir os materiais e as espessuras da alvenaria.

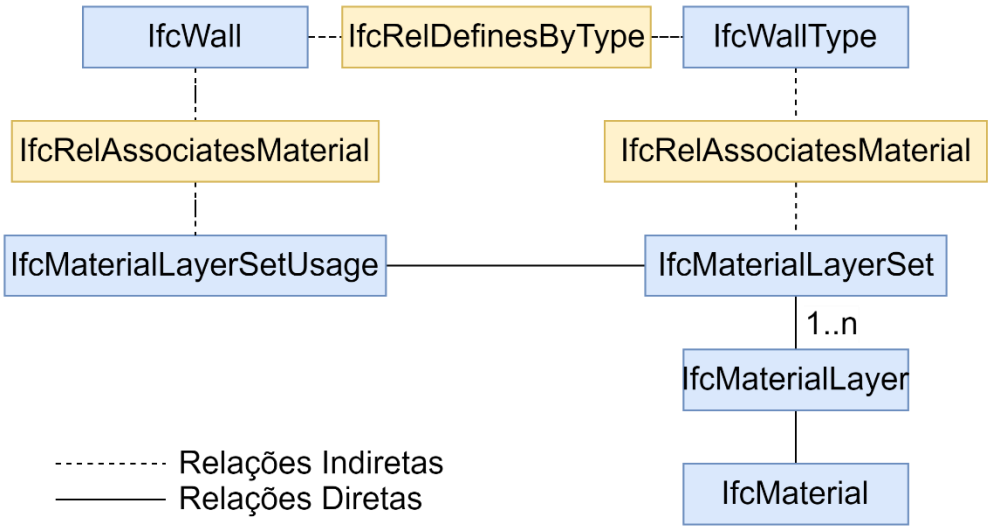
Figura 2: Esquema de classes IFC para definição de representações.



Fonte: o autor.

Ao instanciar um tipo de parede (*IfcWallType*), o elemento define como essas camadas serão posicionadas em relação ao eixo de locação (*IfcMaterialLayerSetUsage*). Por sua vez, a geometria 3D da parede é gerada computacionalmente pela extrusão de um perfil perpendicularmente ao plano do eixo (*IfcExtrudedAreaSolid*).

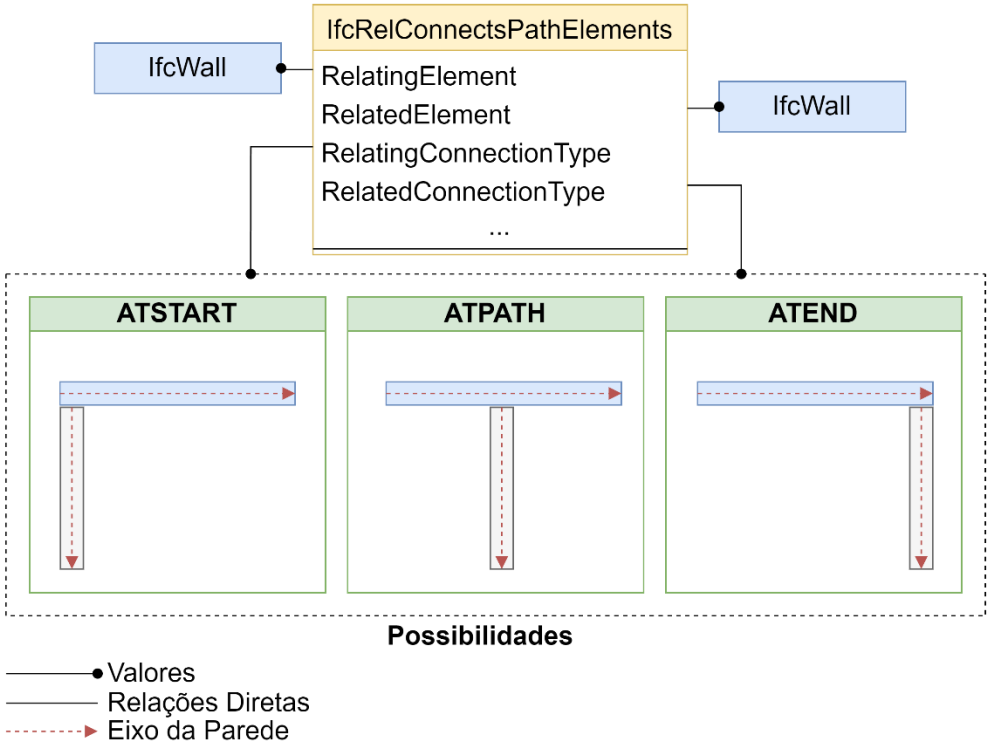
Figura 3: Esquema de classes IFC para a definição de camadas de material.



Fonte: o autor.

Além das propriedades geométricas, a representação *Axis* fornece informações cruciais sobre a conectividade entre os elementos. A intersecção das paredes é garantida por entidades de conectividade de eixos (*IfcRelConnectsPathElements*), que definem o tipo de junção geométrica entre as paredes conectadas.

Figura 4: Esquema de classes IFC para definição de conectividade entre elementos com eixo.

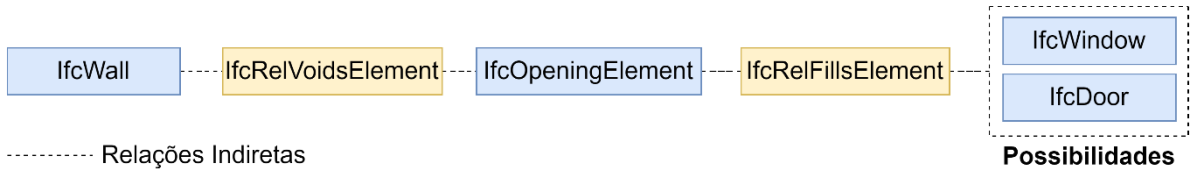


Fonte: o autor.

Considerando as exigências analíticas do MPE, a extração da representação *Axis* associada às propriedades de materiais fornece todas as coordenadas e parâmetros necessários para interpretar a posição e a relação estrutural entre os painéis. Sendo assim, este é o único requisito estrito de exportação exigido para a aplicação do script.

Por fim, a modelagem de vãos acompanha a lógica nativa do padrão IFC. As aberturas interagem com a parede hospedeira através de uma relação de subtração de volume (*IfcRelVoidsElement*), criando um vazio que é posteriormente preenchido por componentes como portas ou janelas (*IfcRelFillsElement*). A hierarquia padrão atende à leitura do algoritmo, dispensando rotinas excepcionais para reconhecer descontinuidades na alvenaria.

Figura 5: Esquema de classes IFC para definição de aberturas.



Fonte: o autor.

O DOMÍNIO ANALÍTICO NO PADRÃO IFC

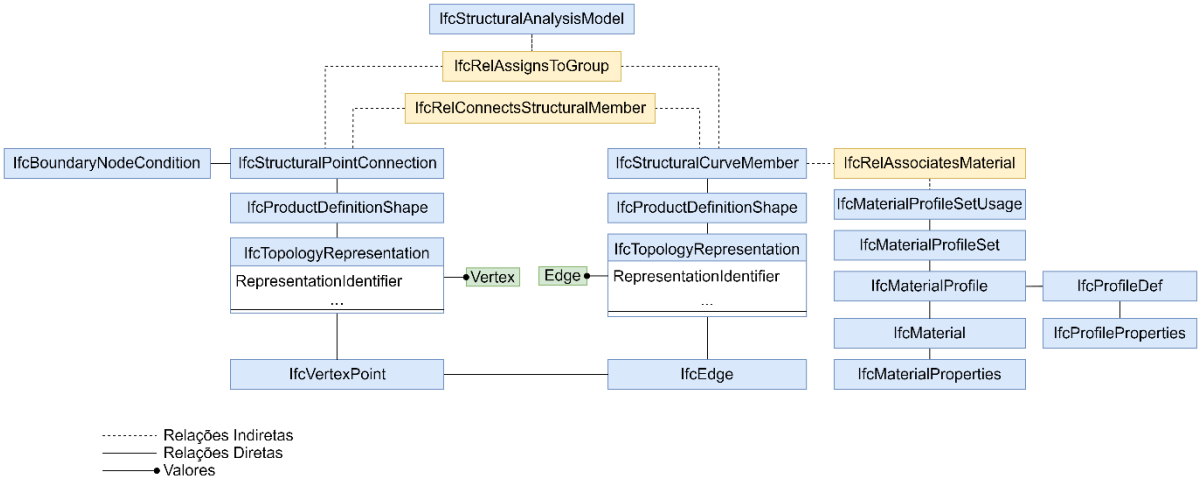
Diferente dos domínios geométricos dos elementos (*IfcArchitecturalDomain* e *IfcStructuralElementsDomain*), focados na representação física e construtiva da peça, o domínio analítico (*IfcStructuralAnalysisDomain*) concentra-se na topologia, propriedades mecânicas e condições de contorno. A classe *IfcStructuralAnalysisModel* atua como o contêiner do modelo analítico, agrupando os dados da análise estrutural.

No contexto do MPE, o algoritmo utiliza duas entidades fundamentais:

- ***IfcStructuralPointConnection*:** Representa os nós analíticos do sistema. Geometricamente, são definidos como pontos singulares no espaço tridimensional. No fluxo desenvolvido, esses nós são gerados nas intersecções dos eixos das paredes, nas extremidades dos lintéis e na base do pavimento. É também nesta entidade que as condições de contorno são aplicadas.
- ***IfcStructuralCurveMember*:** Representa os elementos de barra do pórtico. Esta classe modela os pilares equivalentes e os lintéis. As barras rígidas também são instâncias dessa classe e com comportamento rígido garantido pela atribuição de propriedades mecânicas de rigidez infinita.

A conectividade entre elementos é descrita através da classe de relação *IfcRelConnectsStructuralMember*. Esta entidade vincula uma barra aos seus nós de extremidade, definindo os graus de liberdade fundamentais para o comportamento solidário da alvenaria estrutural.

Figura 6: Esquema de classes IFC para definição de modelos analíticos reticulados



Fonte: o autor.

MÉTODO

Esta pesquisa adota o método *Design Science Research*, focado no desenvolvimento de artefatos para a resolução de problemas práticos. O delineamento da pesquisa compreendeu a definição de requisitos baseados no padrão IFC 4/4.3 e a construção de um artefato computacional (algoritmo) para automatizar a geração do modelo analítico. A avaliação da utilidade e eficácia do artefato foi realizada por meio de um estudo de caso em uma edificação de 3.575 m², onde os resultados foram validados mediante auditoria paramétrica e visual no software BIMCollab Zoom, assegurando a integridade da malha nodal e a precisão das seções transversais em conformidade com o modelo físico original.

O ALGORITMO PROPOSTO: FLUXO COMPUTACIONAL

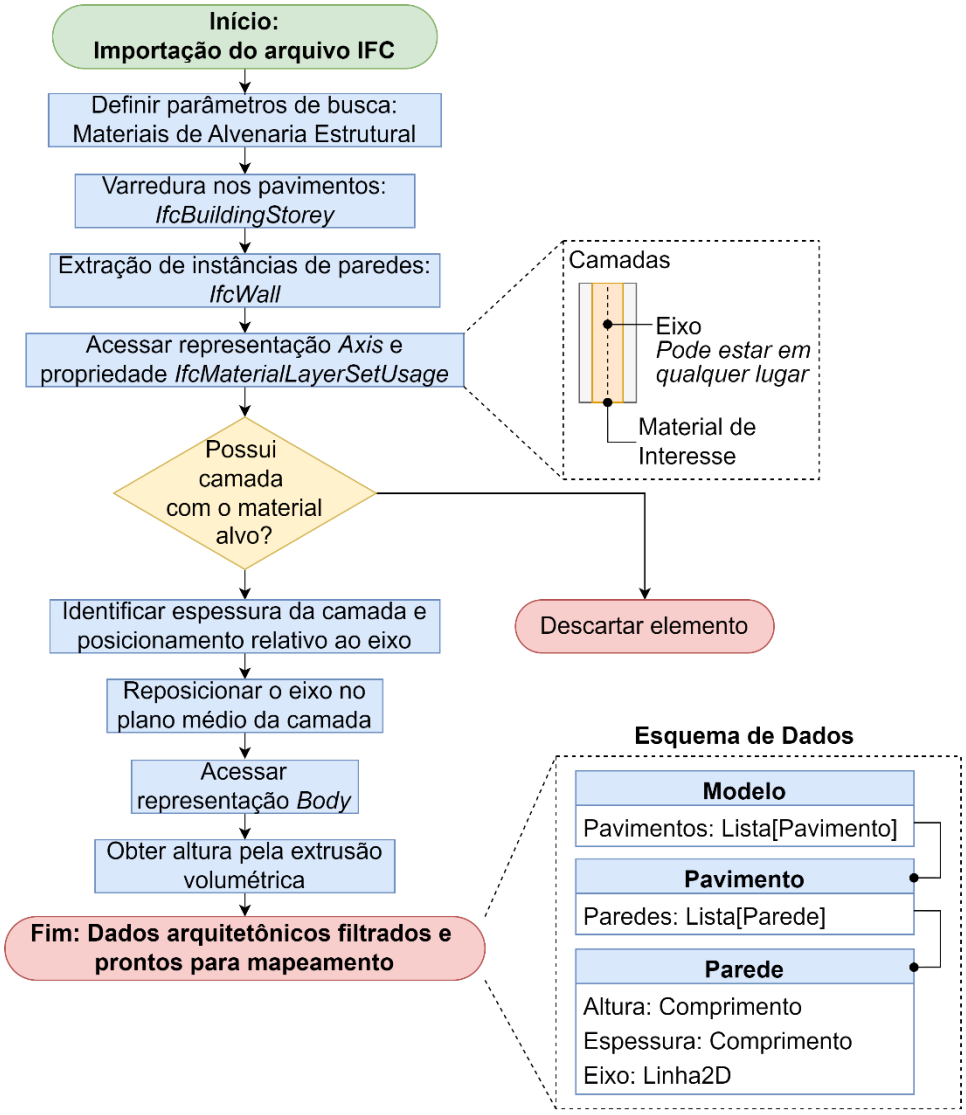
O artefato foi desenvolvido em linguagem Python, utilizando a biblioteca *open-source* IfcOpenShell para manipular as entidades IFC. A estruturação lógica e otimização das funções de varredura geométrica contaram com o suporte de inteligência artificial generativa [20]. A rotina opera sequencialmente em quatro etapas lógicas que transformam o modelo arquitetônico no domínio de análise estrutural:

LEITURA E FILTRAGEM DE DADOS ARQUITETÔNICOS

O fluxo inicia com a importação do arquivo IFC. A partir do nome dos materiais que devem ser interpretados como alvenaria estrutural, o script realiza uma varredura nos pavimentos (*IfcBuildingStorey*) e extrai as instâncias de paredes (*IfcWall*). O algoritmo acessa a representação *Axis* e as propriedades de materiais (*IfcMaterialLayerSetUsage*) para identificar a espessura total do painel e seu posicionamento, descartando camadas não especificadas ou paredes que não as possuem, evitando a inclusão indevida de vedações. Um aspecto crítico dessa etapa é o reposicionamento do eixo de referência para o plano médio da camada de alvenaria:

como o *IfcMaterialLayerSetUsage* pode posicionar o eixo original em qualquer ponto da seção transversal, dependendo de como o modelo arquitetônico foi construído, essa correção é indispensável para assegurar que o pilar equivalente, gerado nas etapas seguintes, coincida com o centroide real da parede. Adicionalmente, as alturas das paredes são obtidas pela extrusão da sua representação *Body*, dispensando o processamento de sua geometria volumétrica completa.

Figura 7: Fluxo de leitura e filtragem de dados arquitetônicos.



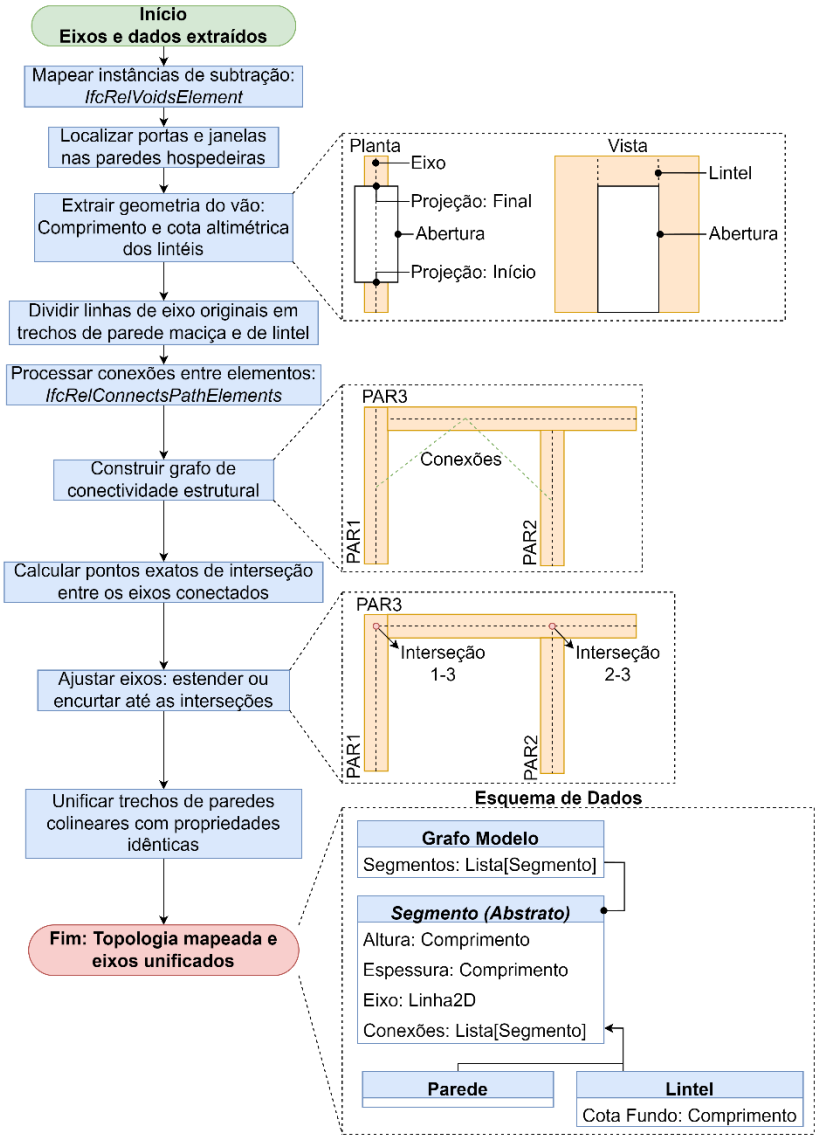
Fonte: o autor.

MAPEAMENTO TOPOLÓGICO E TRATAMENTO DE VÃOS

Com as linhas de eixo extraídas, o código mapeia as instâncias de subtração (*IfcRelVoidsElement*) para localizar portas e janelas. É nesta etapa que a geometria do vão define o comprimento e a cota altimétrica dos lintéis. A partir dessa leitura, o script divide os eixos em trechos de parede maciça e de lintel, reproduzindo a lógica física do MPE: cada abertura interrompe a continuidade do pilar equivalente e introduz um segmento horizontal de conexão.

Em seguida, o algoritmo processa as conexões entre os elementos (*IfcRelConnectsPathElements*), construindo um grafo de conectividade entre as paredes para calcular os pontos de interseção entre os eixos que, se necessário, são estendidos ou encurtados. Essa correção é necessária porque os eixos de paredes adjacentes frequentemente não se tocam exatamente ou se sobrepõem de forma inconsistente, uma limitação comum que, se não tratada, geraria nós desconectados ou duplicados no modelo analítico. Por fim, o algoritmo unifica trechos de paredes colineares com propriedades idênticas, garantindo a representação do comportamento estrutural contínuo do painel e evitando a fragmentação artificial de um mesmo elemento portante em múltiplos segmentos analíticos.

Figura 8: Fluxo de mapeamento topológico



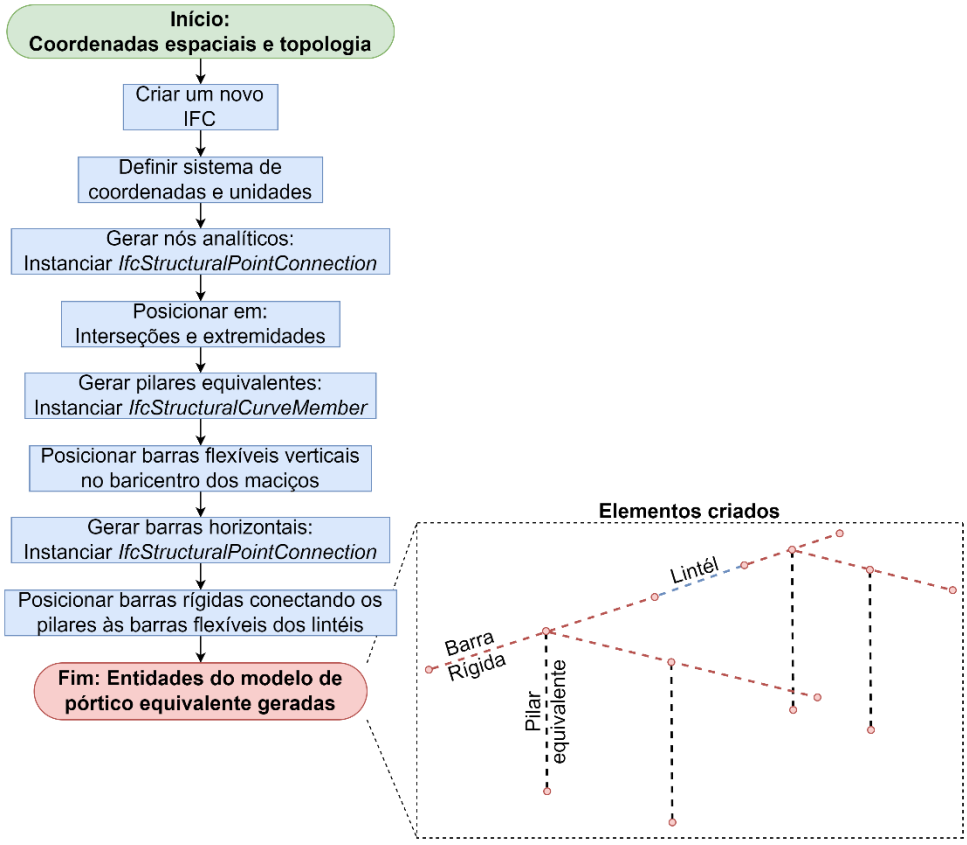
Fonte: o autor.

INSTANCIAÇÃO DO DOMÍNIO ANALÍTICO

Após calcular as coordenadas espaciais e a topologia necessárias, o algoritmo cria um arquivo IFC e define o sistema de coordenadas e as unidades do modelo analítico, isolando o domínio *IfcStructuralAnalysisDomain* do arquivo arquitetônico original.

Em seguida, são instanciados os nós (*IfcStructuralPointConnection*) nas intersecções dos eixos, nas extremidades dos lintéis e na base do pavimento. Os pilares equivalentes (*IfcStructuralCurveMember*) são então gerados como barras flexíveis verticais posicionadas no eixo centroidal de cada trecho maciço, enquanto os lintéis são representados por barras horizontais análogas sobre os vãos. Por fim, barras rígidas, com rigidez mecânica teórica infinita, conectam os nós baricêntricos aos nós dos lintéis, reproduzindo a excentricidade real entre o eixo da parede e a face da abertura, conforme discutido na seção sobre o MPE.

Figura 9: Fluxo de instanciação do domínio analítico.



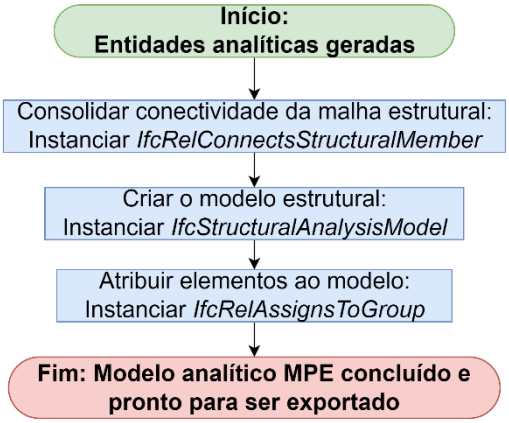
Fonte: o autor.

CONECTIVIDADE E EXPORTAÇÃO

A última etapa consolida a conectividade da malha. O algoritmo utiliza a *IfcRelConnectsStructuralMember* para declarar quais barras estão conectadas a quais nós, garantindo a continuidade mecânica. Sem essa declaração explícita, os elementos gerados permaneceriam geometricamente coincidentes, mas topologicamente desconectados para fins de cálculo. Todos os elementos analíticos gerados são então agrupados sob a *IfcStructuralAnalysisModel*, que atua como contêiner padronizado do

domínio estrutural. O resultado é exportado como um novo arquivo IFC independente, contendo exclusivamente a malha reticulada estrutural, o que permite sua importação direta em softwares de análise compatíveis com o *IfcStructuralAnalysisDomain*.

Figura 10: Fluxo de conectividade e exportação.

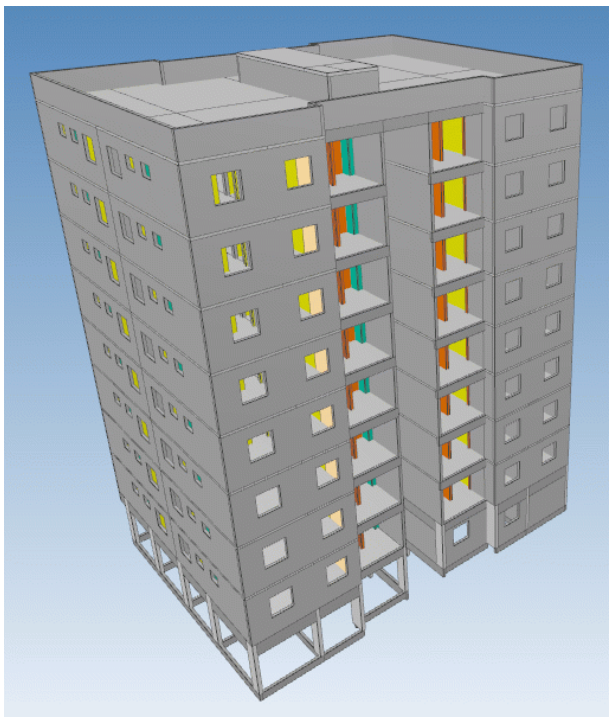


Fonte: o autor.

RESULTADOS E DISCUSSÕES

Para a etapa de demonstração e avaliação do artefato, a rotina computacional foi aplicada a um modelo arquitetônico de referência (Figura 11), exportado no formato IFC4. O estudo de caso consistiu em uma edificação em alvenaria estrutural com 242 painéis portantes e 478 aberturas, totalizando uma área construída de 3.575 m².

Figura 11: Modelo IFC de referência

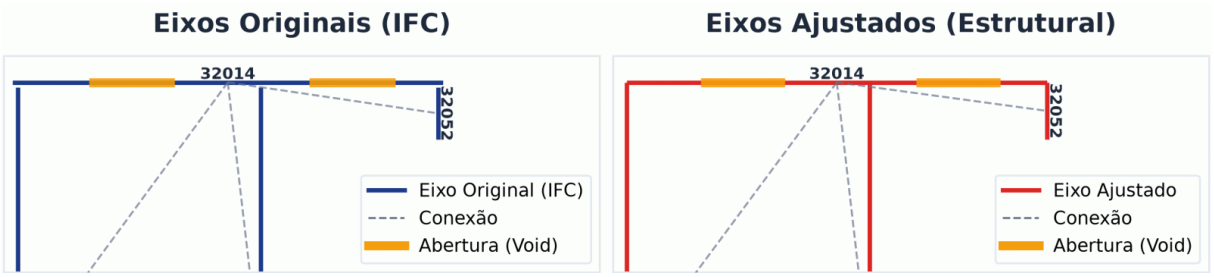


Fonte: o autor.

O tempo total de processamento — compreendendo a leitura volumétrica, extração de eixos, cálculo de interseções, geração das entidades do MPE e exportação final — foi de 20,52 segundos. Este resultado evidencia a alta eficiência computacional do artefato, especialmente quando comparado ao tempo demandado para o lançamento manual ou conversão assistida em softwares comerciais.

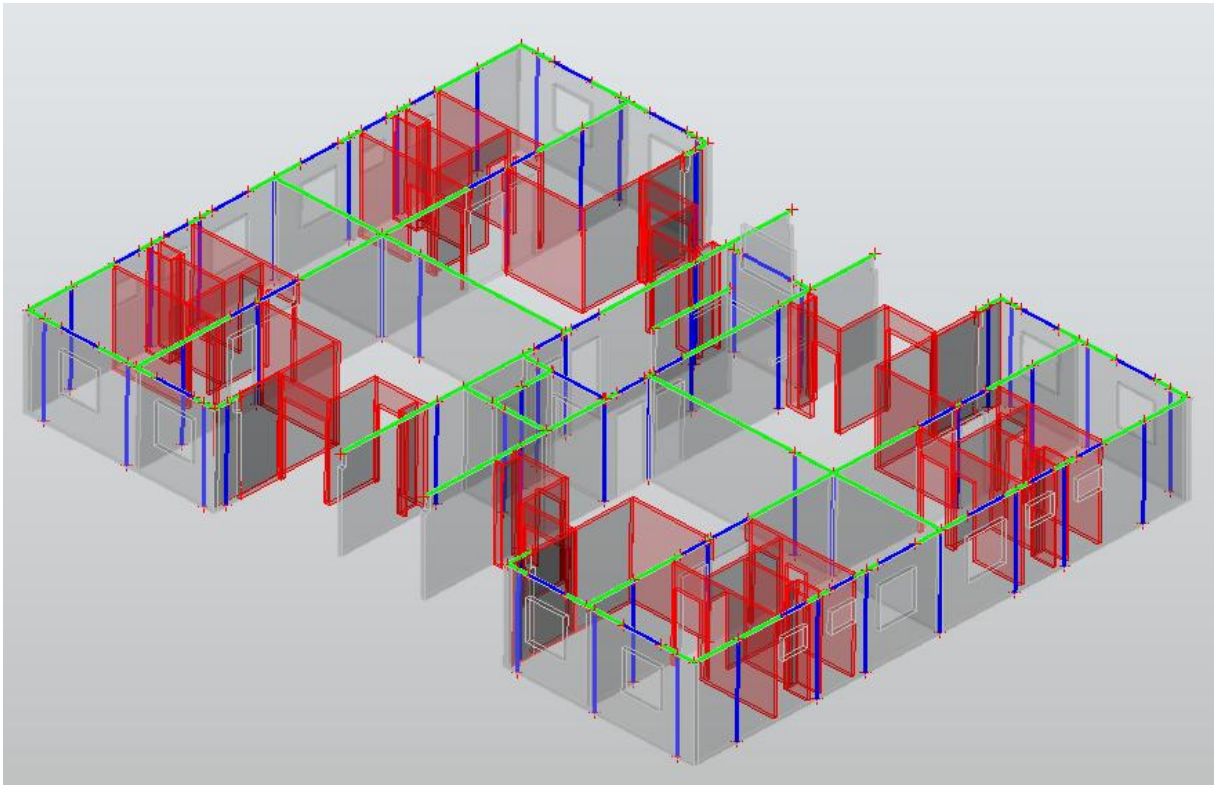
A Figura 12 apresenta um trecho do pavimento tipo onde é possível observar o ajuste realizado pelo algoritmo: os eixos originais da arquitetura foram processados para garantir a continuidade física necessária ao modelo estrutural. O algoritmo fragmentou corretamente as regiões adjacentes às aberturas e gerou os lintéis horizontais, preservando a lógica estática do MPE

Figura 12: Trecho do pavimento tipo apresentando eixos originais e eixos ajustados.



Fonte: o autor.

Figura 13: Pavimento tipo convertido e com a representação da arquitetura transparente.

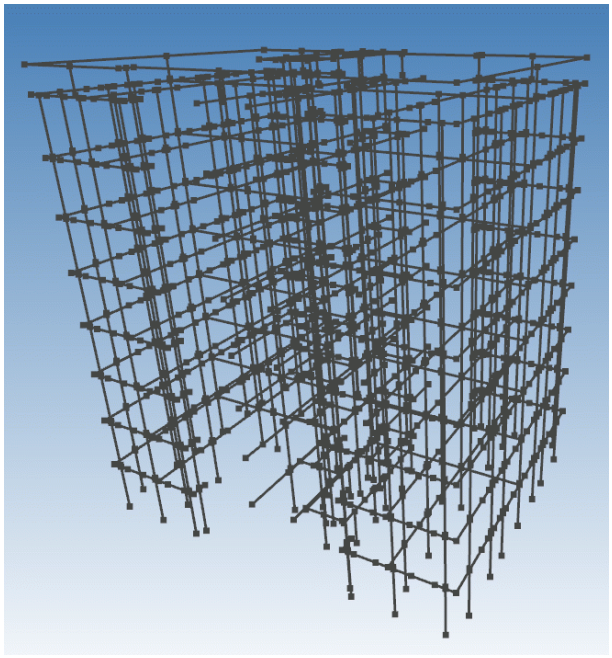


Nota: Em vermelho as paredes ignoradas. Fonte: o autor.

A validação qualitativa ocorreu no visualizador BIMCollab Zoom, onde a importação do *IfcStructuralAnalysisModel* demonstrou ausência de perdas de conectividade; A

Figura 14 apresenta a topologia unifilar gerada. O algoritmo fragmentou corretamente as regiões adjacentes às aberturas e gerou os lintéis horizontais, preservando a lógica estática do MPE. Na validação paramétrica, confirmou-se o enriquecimento semântico. As barras topológicas assimilaram a espessura de 14 cm do painel original, convertendo esse dado geométrico em seções transversais ativas para o cálculo da matriz de rigidez.

Figura 14: Modelo IFC estrutural.



Fonte: o autor.

Apesar da eliminação da barreira de interoperabilidade inicial, o método detém limitações. A atribuição de cargas verticais e ações horizontais (vento) ainda demanda intervenção manual no ambiente de cálculo. Contudo, a integridade da malha gerada válida a premissa de que o padrão openBIM é um repositório maduro para sustentar integrações diretas entre arquitetura e engenharia.

CONCLUSÃO

Este trabalho validou um método *open-source* para converter modelos físicos de alvenaria estrutural em modelos analíticos do tipo Pórtico Equivalente. Através da biblioteca IfcOpenShell em Python, demonstrou-se a viabilidade de contornar a complexidade geométrica da redução de dimensionalidade de volumes 3D. A extração direta de diretrizes nativas (*Axis*) e a leitura de relacionamentos topológicos (*IfcRelConnectsPathElements* e *IfcRelVoidsElement*) são suficientes para deduzir eixos, interrupções, lintéis e nós.

A transição para o domínio *IfcStructuralAnalysisModel* e a validação da malha atestam que o padrão openBIM possui maturidade semântica para sustentar a interoperabilidade fluida na representação MPE, reduzindo a dependência de conversores proprietários.

Como trabalhos futuros, planeja-se a expansão do algoritmo para englobar o mapeamento de cargas gravitacionais e horizontais, além da integração direta do script com motores de cálculo de elementos finitos via API, estabelecendo um fluxo de trabalho em ciclo fechado para retroalimentar o modelo IFC federado de forma nativa e contínua.

REFERÊNCIAS

[1] G. Mohamad, *Construções em alvenaria estrutural: materiais, projeto e desempenho*, 3rd ed. Editora Edgard Blücher Ltda., 2020.

[2] R. Sacks, C. Eastman, G. Lee, and P. Teicholz, *BIM Handbook: A Guide to Building Information Modeling for Owners, Designers, Engineers, Contractors, and Facility Managers*, 3rd ed. Hoboken, New Jersey: John Wiley & Sons, 2018.

[3] X. Y. Deng and T.-Y. P. Chang, "Creating structural model from IFC-Based architectural model," in *Joint International Conference on Computing and Decision Making in Civil and Building Engineering*, Montreal, Canadá, 2006, pp. 3687–3695.

[4] L. Qin, X. Deng, and X. Liu, "Industry Foundation Classes Based Integration of Architectural Design and Structural Analysis," *J. Shanghai Jiaotong Univ. Sci.*, vol. 16, no. 1, pp. 83–90, 2011.

[5] Z.-Z. Hu, X.-Y. Zhang, H.-W. Wang, and M. Kassem, "Improving interoperability between architectural and structural design models: An industry foundation classes-based approach with web-based tools," *Autom. Constr.*, vol. 66, pp. 29–42, 2016.

[6] G. Sibenik, I. Kovacic, V. Petrinas, and W. Sprenger, "Implementation of Open Data Exchange between Architectural Design and Structural Analysis Models," *Buildings*, vol. 11, no. 605, 2021, doi: 10.3390/buildings11120605.

[7] Z. Xu and others, "Developing an Extended IFC Data Schema and Mesh Generation Framework for Finite Element Modeling," *Advances in Civil Engineering*, vol. 2019, pp. 1–19, 2019.

[8] J. Jia and others, "An Automatic Generation Method of Finite Element Model Based on BIM and Ontology," *Buildings*, vol. 12, no. 11, p. 1949, 2022.

[9] L. Xiang, G. Li, and H. Li, "Automatic generation of Structural models from BIM models using Semantics and Machine Learning," 2023.

[10] J. V. Silva and C. C. Nunes, "Comparação de deslocamentos e tensões normais de um grupo de paredes de um edifício em alvenaria estrutural: método dos elementos finitos versus elementos de pórtico espacial," *E&S - Engineering and Science*, vol. 8, no. 3, pp. 21–36, 2019, doi: 10.18607/ES201988588.

[11] AltoQi Tecnologia Aplicada à Engenharia, "Eberick: Software para projeto estrutural de edificações," 2026, *Florianópolis, SC*. [Online]. Available: https://www.altoqi.com.br/produtos/eberick

[12] TQS Informática Ltda., "Sistema TQS: Softwares para projeto de estruturas de concreto armado," 2026, *São Paulo, SP*. [Online]. Available: https://www.tqs.com.br/

[13] Autodesk Inc., "Robot Structural Analysis Professional," 2026, *San Rafael, CA*. [Online]. Available: https://www.autodesk.com/products/robot-structural-analysis/overview

[14] Dlubal Software GmbH, "RFEM: Software de análise estrutural por elementos finitos," 2026, *Tiefenbach, Alemanha*. [Online]. Available: https://www.dlubal.com/pt/produtos/software-aef-rfem/o-que-e-o-rfem

[15] T. Krijnen and IfcOpenShell Contributors, "IfcOpenShell: The open source IFC toolkit and geometry engine," 2026. [Online]. Available: https://ifcopenshell.org/

[16] K. Palácio, “Desenvolvimento e implementação de um software para análise tridimensional de edif\’\icios altos em alvenaria estrutural,” Escola de Engenharia, Universidade Federal de Minas Gerais, Belo Horizonte, 2001.

[17] buildingSMART International, “Model View Definition (MVD),” 2024. [Online]. Available: https://technical.buildingsmart.org/standards/ifc/mvd/

[18] buildingSMART International, “IFC 4.3 Documentation: Fundamental Concepts and Assumptions,” 2024. [Online]. Available: https://ifc43-docs.standards.buildingsmart.org/IFC/RELEASE/IFC4x3/HTML/concepts/content.html

[19] buildingSMART International, “IFC4 Reference View 1.2 (MVD),” 2017. [Online]. Available: https://standards.buildingsmart.org/MVD/RELEASE/IFC4/ADD2_TC1/RV1_2/HTML/

[20] Google LLC, “Gemini 3: Modelo de inteligência artificial generativa,” 2026, *Mountain View, CA*. [Online]. Available: https://gemini.google.com/