

# Automatização do processo de licenciamento – Metodologia e caso de estudo

<https://doi.org/10.21814/uminho.ed.77.13>

Nycolas Glerean<sup>1</sup>, Alaa Alhariri<sup>1</sup>,  
Ana Rocha<sup>1</sup>, Luís Sanhudo<sup>1</sup>, António Aguiar Costa<sup>1</sup>

<sup>1</sup> BUILT CoLAB – Collaborative Laboratory for the Future Built Environment, Portugal

## Resumo

Com os avanços tecnológicos dos últimos anos, o *software* de validação automática *Building Information Modelling* (BIM) tem demonstrado ser uma eficaz solução para a otimização de processos do setor da Construção, nomeadamente no contexto de licenciamento. Este tipo de solução visa reduzir a duração e o erro humano associado a este processo, bem como melhorar a colaboração entre entidades, produzindo um impacto positivo no setor da Construção em termos económicos, de sustentabilidade e de segurança.

Tendo em vista a crescente adoção destas ferramentas, o presente artigo introduz uma metodologia que visa apoiar e automatizar o processo de licenciamento a nível municipal e nacional. Esta metodologia inclui três fases distintas: (1) análise e tradução de regulamentos para linguagem digital; (2) classificação de zonas e introdução de parâmetros em ambiente de modelação; (3) execução do processo de validação e visualização de resultados. O potencial desta abordagem é ilustrado por meio de um caso de estudo, no qual é apresentada uma versão preliminar de um *software* de *code-checking* capaz de analisar e validar um modelo BIM em conformidade com parte do regulamento nacional de acessibilidade.

## 1. Introdução

Os recentes esforços de digitalização do setor da Arquitetura, Engenharia e Construção (AEC) têm proporcionado novas oportunidades para a criação de inovadores fluxos de trabalho, nomeadamente pela automatização de processos tradicionalmente manuais. Entre estes, a literatura aponta para a validação automática de modelos BIM de acordo com normas, regulamentos e requisitos da Construção, como uma boa abordagem para potenciar a aceleração e melhoria da qualidade do processo de licenciamento, gerando um impacto positivo para o setor ao nível da sua economia, sustentabilidade e segurança [1].

Contudo, apesar da identificação destas vantagens por parte da literatura, existe ainda uma falha no desenvolvimento de uma ferramenta que possibilite a automatização do processo de licenciamento, frequentemente referido de *e-permitting* [2].

Assim, apesar da automatização integral deste processo não ser atualmente um objetivo verosímil, nem desejado, o presente artigo propõe uma metodologia para uniformização da interpretação de regulamentos e modelação BIM, criando um caminho a seguir para que se possa atingir, num futuro próximo, esta automatização a nível nacional. Para validação da metodologia, o artigo apresenta ainda uma versão preliminar de um *software* de *code-checking*, capaz de examinar e validar um modelo BIM em conformidade com parte do regulamento nacional de acessibilidade. Este *software* é constituído por duas ferramentas distintas: *BIM Classification* e *BIM Checker*.

Para além desta secção introdutória o presente artigo encontra-se dividido em seis secções. A Secção 2 examina o estado da arte associado ao tópico de *code-checking* e *e-permitting*, abordando ainda as ferramentas que inspiraram o *software* proposto. A Secção 3 delinea a metodologia proposta. A Secção 4 introduz o *software* desenvolvido com base na metodologia apresentada. A Secção 5 apresenta a validação a metodologia e do *software* proposto com base num pequeno caso de estudo. E, por fim, a Secção 6, apresenta as principais conclusões, bem como os trabalhos futuros já planeados.

## 2. Estado da Arte

Segundo Shahi et al. [3], os sistemas de licenciamento podem ser classificados em quatro níveis distintos, consoante o seu nível de digitalização e automatização. Nomeadamente:

- Nível 0 – Sistema de licenciamento tradicional, suportado por documentação em papel e sem recurso a processos de automatização;
- Nível 1 – Sistema de licenciamento suportado por documentação digital sobre a forma de ficheiros CAD (*Computer-Aided Design*), mas sem recurso a processos de automatização. Atualmente, é o sistema de licenciamento mais comum;

- Nível 2 – Sistema de licenciamento suportado por documentação digital sobre a forma de ficheiros BIM, com múltiplos mecanismos de automatização que aceleram o processo de licenciamento e diminuem o erro-humano associado ao mesmo;
- Nível 3 – Sistema de licenciamento suportado por documentação digital sobre a forma de ficheiros BIM e SIG (Sistema de Informação Geográfica) integrados, capturando coordenadas geodésicas, informação espacial e relações entre objetos, permitindo a completa automatização do processo de licenciamento.

Em [3], os autores expandem cada um destes níveis com múltiplos subníveis adicionais, identificando ainda que a maioria dos processos de licenciamento na literatura representam sistemas de nível 1, com a ambição de transitarem para o nível 2. Neste sentido, existe a necessidade de desenvolver sistemas de *e-permitting* (incluídos nos níveis 2 e 3), descritos na literatura como sistemas que visam a digitalização e automatização total ou parcial do processo de licenciamento tradicional. Esta automatização é suportada por *software* de *code-checking*, que verifica, de forma automática, o modelo BIM conforme regras previamente estabelecidas, emitindo posteriormente os resultados da análise efetuada. Segundo [4], este *software* é tipicamente constituído por quatro componentes:

- Tradução de regras para linguagem legível por computador;
- Preparação do conteúdo do modelo BIM;
- Execução do processo de validação;
- Preparação e representação dos resultados obtidos.

No que diz respeito a *software* de *code-checking* existente, uma das primeiras ferramentas com este propósito surgiu na plataforma CORENET (*Construction and Real Estate Network*), desenvolvida pela *Building Construction Authority*, em 1995. Esta suporta um serviço *web* para verificação técnica de regulamentos de edifícios, do qual faz parte o *software e-PlanCheck* [5]. Este é um *software* de *code-checking*, constituído por três vertentes: *e-submission*, *e-PlanCheck* e *e-info*. O *e-PlanCheck* foi testado em projetos piloto na Noruega e nos Estados Unidos da América, sendo que atualmente cobre ~92% das normas associadas ao *Integrated Building Plan* e ~77% do *Integrated Building Service*, em Singapura [2, 6, 7].

Como outro exemplo deste tipo de *software*, o *Solibri Model Checker* (SMC) [8] é uma das ferramentas mais populares para *code-checking* e avaliação da qualidade de modelos BIM. Este *software* foi criado pela *Solibri*, em 2000, tendo por base o modelo IFC (*Industry Foundation Classes*) [9]. A estrutura de verificações do SMC é baseada em regras paramétricas, classificando os resultados obtidos consoante a sua gravidade (crítico; moderado; e baixa gravidade) [10].

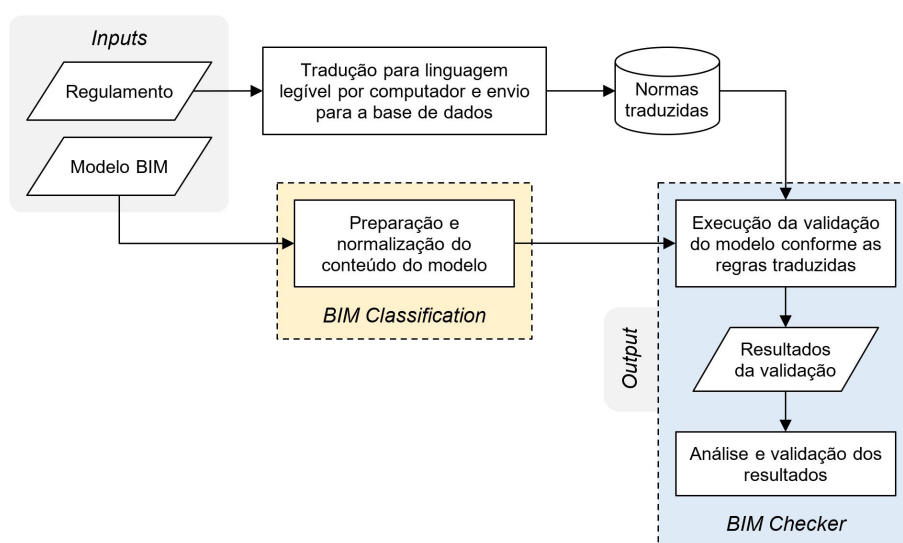
Noutro exemplo, o *Model Checker* [11] da *Autodesk* permite uma rápida configuração e análise de regras paramétricas personalizadas pelo utilizador. Para desenvolvimento destas regras o *Model Checker* vem acompanhado do *Model Checker Configurator*, que permite uma fácil criação e edição de regras BIM.

Outros exemplos de relevo incluem [2, 4, 12]: *DesignCheck*; *Statsbygg*; *Jotne EDModelChecker*; e *SMARTcodes*. De realçar a importância deste último exemplo, que teve como foco a tradução de linguagem natural em linguagem legível por computador, permitindo a tradução de termos existentes em regulamentos para propriedades BIM, nomeadamente, propriedades incluídas no formato IFC.

### 3. Metodologia

A Figura 1 apresenta a metodologia proposta. Esta inclui os processos principais de *code-checking* introduzidos por Eastman et al. [4] e referidos na Secção 2. A estas componentes acresce ainda os *inputs* da metodologia, nomeadamente: o regulamento e o modelo BIM a validar. Em termos de fluxo de trabalho, o processo pode ser dividido em três vertentes:

- Numa fase inicial, o regulamento seleccionado passa por um processo de análise e tradução para linguagem legível por computador, culminando num conjunto de regras que são introduzidas numa base de dados.
- Paralelamente a este processo, o modelo BIM introduzido passa por um processo de preparação, no qual a sua informação é uniformizada para ser analisada face às regras geradas – este processo de uniformização é realizado com o auxílio da ferramenta *BIM Classification*, que integra o *software* desenvolvido para este propósito.
- Por fim, a base de dados gerada e o modelo tratado conectam à ferramenta *BIM Checker*, (também parte integrante do *software* desenvolvido) que permite executar, de forma automática, a validação do modelo BIM conforme as regras geradas. Terminado o processo de validação, a mesma ferramenta possibilita a análise e visualização dos resultados.



**Figura 1**  
Metodologia proposta.

Cada uma destas vertentes é explorada, em maior detalhe, na Secção 4. Para desenvolvimento da base de dados foi utilizada a linguagem de programação *SQL*.

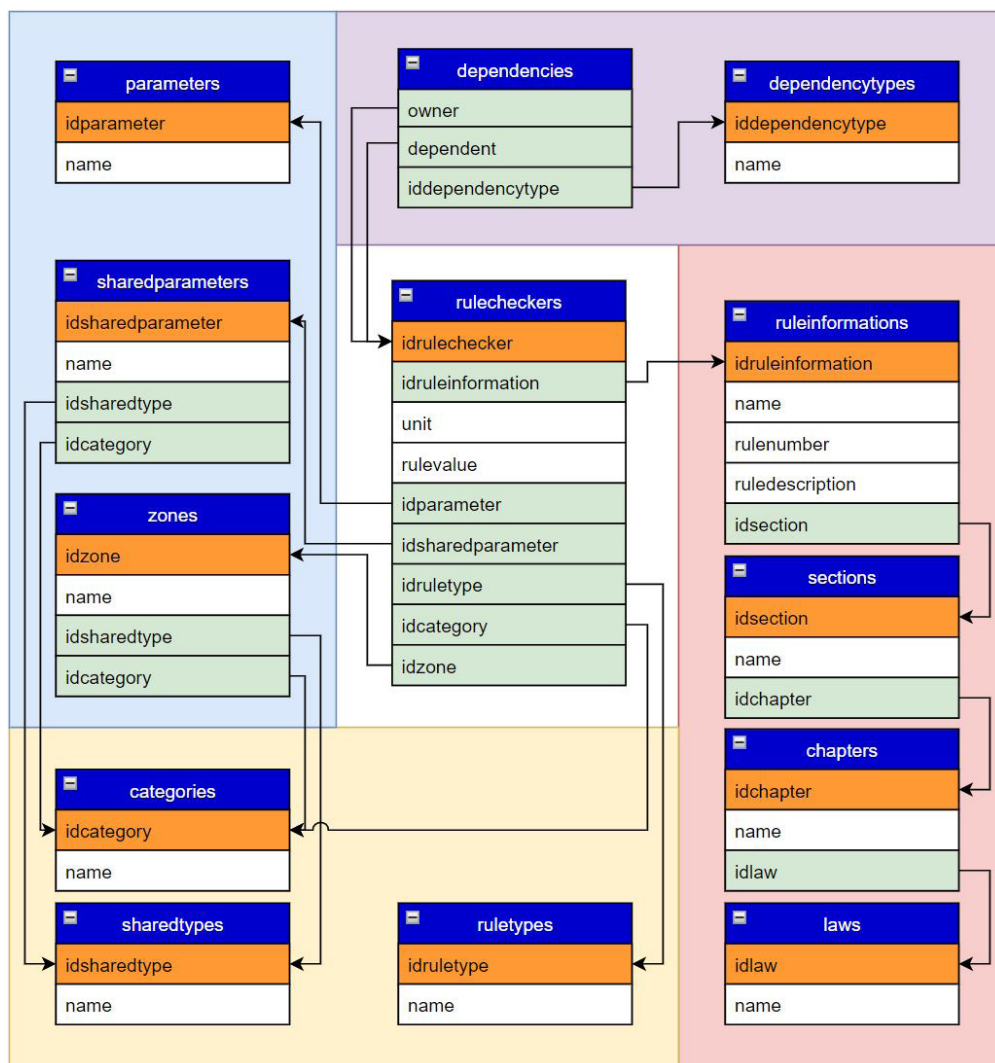
O *software*, composto pelas ferramentas *BIM Classification* e *BIM Checker*, foi desenvolvido com recurso ao *Windows Presentation Foundation* (WPF), uma plataforma gráfica *open-source* desenvolvida pela *Microsoft*. Para a base do código foi utilizado C# e para a componente gráfica, o *Extensible Application Markup Language* (XAML). Ambas as ferramentas foram desenvolvidas sobre a forma de *plugin* para o *software* de modelação *Autodesk Revit*. Este ambiente de modelação foi escolhido pelo seu elevado número de utilizadores a nível nacional, bem como pela sua *Application Programming Interface* (API) que dispõe de uma forte documentação e conteúdo educativo associado.

## 4. Desenvolvimento

### 4.1. Tradução de Regulamentos

A tradução das normas presentes no regulamento introduzido para linguagem legível por computador é um passo vital ao funcionamento da metodologia. Contudo, esta não é uma tarefa de simples execução, requerendo pelo menos três passos de significativa complexidade: (1) o correto mapeamento da nomenclatura presente no regulamento a parâmetros da API do *software Revit*; (2) a correta utilização de operadores lógicos para tradução das regras estipuladas nas normas do regulamento; e (3) a correta identificação e o estabelecimento de dependências entre normas. Esta tarefa é exacerbada pela ambiguidade existente em múltiplos dos regulamentos, pela constante atualização destes documentos, bem como pelo número e diversidade de normas a serem traduzidas.

Assim, para apoiar a execução desta tarefa, foi formulada a arquitetura da base de dados presente na Figura 2, com o intuito de simplificar e estruturar a abordagem. Esta base de dados serve de repositório às normas traduzidas, estando ligada ao *software BIM Checker* para possibilitar o *code-checking* automático.



**Figura 2**  
Arquitetura da base de dados desenvolvida para armazenamento de regulamentos.

Como visível na Figura 2, a base de dados é composta por quatro grupos de informação, interligados pela tabela *rulecheckers*, no centro da imagem, que alberga toda a informação sobre uma determinada validação. Estes grupos encontram-se coloridos de forma distinta e incluem:

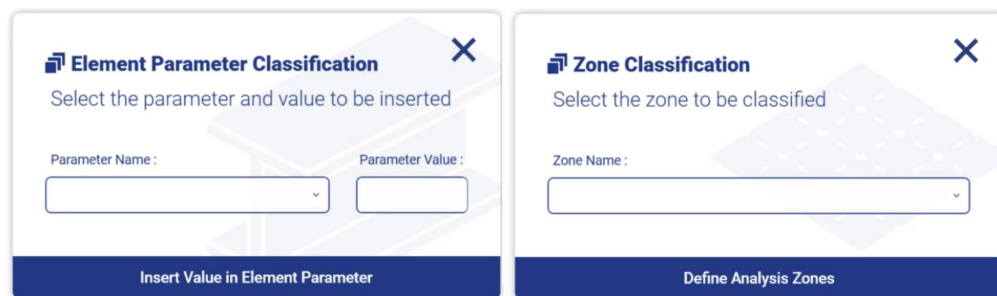
- Informação sobre a norma (grupo a vermelho) associada à validação em questão. Neste grupo encontram-se especificados o nome, número, descrição, secção, capítulo e regulamento da norma;
- Informação sobre dependências (grupo a roxo), onde se estipula uma hierarquia entre diferentes validações, permitindo, por exemplo, que uma validação denominada de *owner* (e respetiva norma associada) só seja aprovada após outras validações, denominadas de *dependents*, também o terem sido (i.e., validação composta).
- Informação sobre os parâmetros (grupo a azul) a serem analisados no *software* de modelação, incluindo parâmetros *default* do *Revit* (tabela *parameters*),

parâmetros introduzidos pelo utilizador (tabela *shared parameters*) e possíveis zonas de análise (tabela *zones*);

- **Informação de suporte** (grupo amarelo), onde é especificada a categoria dos elementos a serem analisados (tabela *categories*; e.g., portas, janelas, quartos), a tipologia do parâmetro a analisar segundo a API do *Revit* (tabela *sharedtype*; e.g., texto, área, sim/não), e o operador lógico da validação (tabela *ruletype*; e.g., maior que, menor que, igual a).

#### 4.2. BIM Classification – Preparação e uniformização do conteúdo do modelo

A preparação e uniformização do modelo BIM para a realização do processo de *code-checking* é um passo fulcral da metodologia proposta. Esta uniformização é realizada a vários níveis, desde a indicação do tipo de parâmetros a introduzir em certos elementos, até à nomenclatura que estes parâmetros devem seguir. Neste sentido, este passo é suportado não só por guias de modelação, mas também pela ferramenta *BIM Classification* (Figura 3), desenvolvida com o intuito de tornar este passo mais intuitivo e expedito. Esta ferramenta permite uma ágil atribuição de parâmetros e zonas a elementos do modelo BIM, incluindo a especificação dos valores associados aos mesmos. Esta atribuição pode ser realizada elemento-a-elemento, para uma maior personalização, ou em simultâneo a múltiplos elementos e/ou categorias.



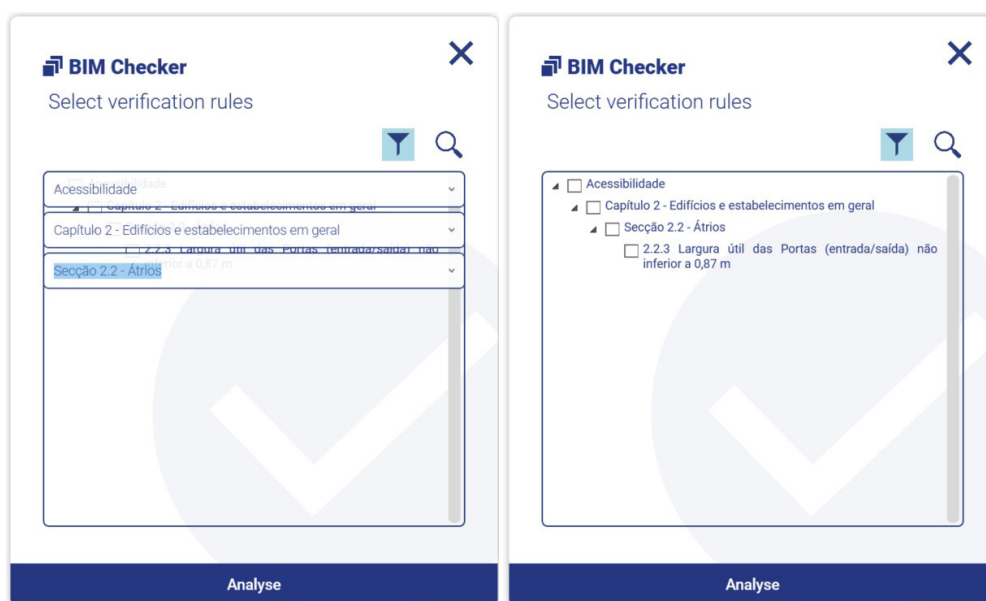
**Figura 3**  
Arquitetura da base de dados desenvolvida para armazenamento de regulamentos.

#### 4.3. BIM Checker

A ferramenta *BIM Checker* permite validar o modelo BIM de forma automática. A ferramenta permite várias validações em simultâneo, analisando todas as normas selecionadas pelo utilizador no painel interativo ilustrado na Figura 4. Para que esta validação ocorra corretamente, todos os parâmetros e zonas devem estar previamente inseridos (Secção 4.2). Caso estes parâmetros não estejam corretamente introduzidos, a validação alerta o utilizador para esse facto. Este alerta é feito por um sistema de cores, que também ilustra o cumprimento/incumprimento da norma:

- Cinzento, caso não existam elementos para validação;
- Amarelo, caso o parâmetro a validar não esteja inserido;
- Vermelho, caso a norma não seja cumprida;
- Verde, caso a norma seja validada.

Os resultados da análise realizada ao modelo BIM são apresentados de duas formas distintas: em formato de texto (na interface do *BIM Checker*) e em formato 3D (diretamente no modelo). No formato de texto é apresentada uma descrição da regra validada, destacando o resultado através do sistema de cores anteriormente descrito. Neste formato são também identificados os elementos validados, ordenados por família. Os resultados podem ser exportados em formato PDF, facilitando a partilha entre intervenientes do projeto. Em relação à visualização no modelo 3D, esta destaca cada um dos elementos com o mesmo sistema de cores, permitindo ao utilizador uma rápida localização dos elementos que necessitam de alterações.



**Figura 4**  
Interface da ferramenta *BIM Checker*. Processo de filtragem de regulamentos (à esquerda) e processo de seleção das normas a validar (à direita).

## 5. Caso de Estudo

### 5.1. Modelo e regras a validar

O presente capítulo apresenta uma validação preliminar da metodologia proposta, bem como do *software* desenvolvido, através da verificação automática de normas técnicas presentes no Decreto-Lei n.º 163/2006 de 8 de agosto [13] – Regime da acessibilidade aos edifícios e estabelecimentos que recebem público, via pública e edifícios habitacionais. Este Decreto-Lei foca a melhoria da acessibilidade de pessoas com mobilidade condicionada, sendo composto por 388 regras, 128 das quais são identificadas por [14] como verificáveis em ambiente BIM.

Destas 128 regras, seleccionou-se para validação, a título de exemplo, a norma 2.2.3, presente na Secção 2.2 – Átrios. Esta indica que “as portas de entrada/saída dos edifícios e estabelecimentos devem ter uma largura útil não inferior a 0,87 m, medida entre a face da folha da porta quando aberta e o batente ou guarnição do lado oposto; se a porta for de batente ou pivotante deve considerar-se a porta na posição



aberta a 90°. A Tabela 1 apresenta a interpretação lógica desta descrição, segundo a API do *Revit*.

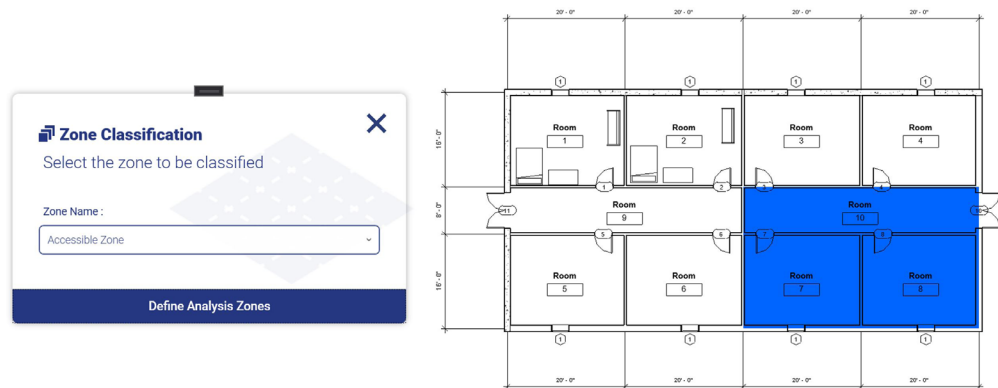
**Tabela 1**

Interpretação lógica da norma 2.2.3, segundo a API do software Autodesk *Revit*.

| Nome  | Parâmetro | Categoria do elemento | Tipo de comparação | Valor de comparação | Zona           |
|---|-----------|-----------------------|--------------------|---------------------|----------------|
| Largura da porta exterior não inferior a 0,87 m | Largura   | Porta exterior        | Maior ou igual     | 0,87                | Zona acessível |

Como é possível observar na Tabela 1, a regra foi traduzida sobre a forma de uma comparação “maior ou igual”, entre o valor do parâmetro “largura” das “portas exteriores” e o valor 0,87 m, identificado na norma. As portas a serem examinadas têm de estar associadas a uma “zona acessível”. Esta zona pode ser criada facilmente utilizando a ferramenta *BIM Classification*, tal como referido na Secção 4.2.

No que diz respeito ao modelo BIM utilizado, optou-se pela criação de um modelo simples, que proporcionasse uma rápida análise dos resultados. O modelo assemelha-se a duas moradias geminadas de apenas um piso, sendo que cada moradia tem quatro quartos interligados por um único corredor que dá acesso à entrada/saída da habitação. A Figura 5 apresenta a planta do modelo desenvolvido, ilustrando ainda a classificação de três zonas do edifício como “zona acessível”, realizada com recurso à ferramenta *BIM Classification*.



**Figura 5**

Vista em planta do modelo criado e classificação de zonas com recurso à ferramenta *BIM Classification*.

## 5.2. Cenários testados e resultado obtidos

Para validar cada um dos possíveis resultados a serem apresentados pela ferramenta *BIM Checker* (Secção 4.3), foram testados quatro cenários distintos: (1) inexistência de portas exteriores a validar na zona classificada de “zona acessível”; (2) inexistência do parâmetro “largura” na porta identificada para validação; (3) porta para validação com um valor do parâmetro “largura” inferior ao requerido pela norma; e (4) porta para validação com um valor do parâmetro “largura” superior ao requerido pela norma.

Tendo em conta estes quatro cenários, a Figura 6 apresenta os resultados obtidos pela ferramenta *BIM Checker*. Como visível nesta figura, todos os cenários apresentaram os resultados pretendidos tanto na vista 3D como na interface do *BIM Checker*. Mais especificamente, o primeiro cenário (Figura 6.A) indica uma falta de elementos (i.e., portas) a verificar – cor cinzenta; o segundo cenário (Figura 6.B) apresenta a falta de parâmetros (i.e., largura) para proceder à sua verificação – cor amarela; o terceiro cenário (Figura 6.C) apresenta um incumprimento da norma – cor vermelha; e, por fim, o quarto cenário (Figura 6.D) apresenta o correto cumprimento da norma – cor verde.



**Figura 6**  
Visualização dos resultados obtido pelo *BIM Checker* nos cenários testados.

## 6. Conclusões e Trabalhos Futuros

Este artigo teve como objetivo disseminar e impulsionar os sistemas de validação automática de modelos BIM, no âmbito do licenciamento automático. Neste sentido, foi proposta uma metodologia para desenvolvimento e implementação deste tipo de soluções, tendo por base a literatura existente. Seguidamente, o potencial desta abordagem foi ilustrado através de um caso de estudo, no qual foi apresentada um *software de code-checking* capaz de analisar e validar um modelo BIM em conformidade com regras de regulamentos nacionais. Este *software* encontra-se dividido em duas ferramentas: *BIM Classification* e *BIM Checker*. A primeira possibilita a uniformização de modelos BIM, enquanto a segunda possibilita a validação dos modelos uniformizados tendo por base normas previamente traduzidas para uma linguagem

legível por computador (i.e., *machine-readable*). O caso de estudo teve resultados positivos, suportando a continuação deste estudo em trabalhos futuros.

Estes trabalhos vão incidir primeiramente na ampliação dos regulamentos abordados, na diversificação das tipologias de regras abrangidas e na criação de uma plataforma digital para disponibilização *online* da ferramenta desenvolvida. Um dos focos desta plataforma passa por permitir a interoperabilidade entre BIM e SIG, permitindo o alcance do nível de 3 de *e-permitting*, por parte da indústria AEC Portuguesa. Estes desenvolvimentos beneficiam já do apoio de múltiplos municípios nacionais.

## 7. Agradecimentos

Este trabalho é cofinanciado pelo Fundo Social Europeu (FSE), através do Programa Operacional Regional do Norte (Norte 2020) e do Programa Operacional Regional de Lisboa (Lisboa 2020) [Referência de Financiamento: NORTE-06-3559-FSE-000176 e LISBOA-05-3559-FSE-000014].

## Referências

- [1] T. H. Beach, J. L. Hippolyte e Y. Rezgui, “Towards the adoption of automated regulatory compliance checking in the built environment”, *Automation in Construction*, 2020.
- [2] H. Narayanswamy, H. Liu e M. Al-Hussein, “BIM-based Automated Design Checking for Building Permit in the Light-Frame Building Industry”, em *Proceedings of the 36<sup>th</sup> International Association for Automation and Robotics in Construction*, Canada, 2019.
- [3] K. Shahi, B. Y. McCabe e A. Shahi, “Framework for Automated Model-Based e-Permitting System for Municipal Jurisdictions”, *Journal of Management in Engineering*, 2019.
- [4] C. Eastman, J. M. Lee, Y. S. Jeong e J. K. Lee, “Automatic rule-based checking of building designs”, *Automation in Construction*, 2009.
- [5] AECbytes, “e-PlanCheck: Singapore’s automated code checking system”, 2005. Disponível em: <https://aecbytes.com/feature/2005/CORENETePlanCheck.html>.
- [6] A. Borrmann, M. König, C. Koch e J. Beetz, *Building information modelling: Technology foundations and industry practice*. Springer International Publishing, 2018.
- [7] Dimyadi, Johannes e Robert Amor. “Automated Building Code Compliance Checking – Where is it at?”, em *19<sup>th</sup> International CIB World Building Congress*, Australia, 2013.
- [8] Solibri, “SMC: Solibri Model Checker”, 2022. Disponível em: <http://solibri.com>.

- [9] V. Bazjanac e D. B. Crawley, “Industry foundation classes and interoperable commercial software in support of design of energy-efficient buildings”, em *Proceedings of Building Simulation’99*, Japan, 1999.
- [10] C. Preidel e A. Borrmann, “Automated Code Compliance Checking Based on a Visual Language and Building Information Modeling”, em *Proceedings of the International Symposium on Automation and Robotics in Construction and Mining*, Finland, 2015.
- [11] Autodesk, “Autodesk BIM Interoperability Tools – Model Checker for Revit”, 2022. Disponível em: <https://interoperability.autodesk.com/modelchecker.php>.
- [12] L. Ding, R. Drogemuller, M. Rosenman e D. Marchant, “Automating code checking for building designs – DesignCheck”, em *Proceedings of the 2<sup>nd</sup> International Conference of the CRC Construction Innovation*, Australia, 2006.
- [13] Decreto-Lei n.º 163/2006 de 8 de agosto do Ministério do Trabalho e da Solidariedade Social. Diário da República: Série I, páginas 5670-5689. Disponível em: <http://dre.pt/dre/>.
- [14] J. P. Rodrigues, “Utilização de modelos BIM para verificação automática de projetos – plano de acessibilidades”, Faculdade de Engenharia da Universidade do Porto, 2015. Disponível em: <https://repositorio-aberto.up.pt/handle/10216/79440>.