

Plataforma BIM para criação de regulamentos digitais no âmbito do licenciamento urbanístico

<https://doi.org/10.21814/uminho.ed.164.13>

**Bruno Muniz¹, José Granja²,
Miguel Azenha³**

¹ Universidade do Minho, ISISE, ARISE, Departamento de Engenharia Civil, Guimarães, Portugal, ORCID:0000-0002-3183-9779

² Universidade do Minho, ISISE, ARISE, Departamento de Engenharia Civil, Guimarães, Portugal, ORCID: 0000-0002-0858-4990

³ Universidade do Minho, ISISE, ARISE, Departamento de Engenharia Civil, Guimarães, Portugal, ORCID:0000-0003-1374-9427

Resumo

Apesar da relevância do licenciamento urbanístico para a garantia da qualidade e desempenho do parque edificado, este é marcado pela baixa utilização de recursos tecnológicos, o que é um dos fatores que implicam na pouca eficiência para a emissão de alvarás de construção e nos atrasos para emissão de licenças de construção. A utilização do BIM para a automatização parcial das verificações de conformidade demonstra ser uma das formas de otimizar o processo de licenciamento, com diversos exemplos de investigação e implementação nas últimas duas décadas. Porém ainda há diversos desafios para se implementar soluções de forma mais abrangente e escalável. Neste contexto, a utilização de sistemas para a criação de regulamentos digitais que possam ser criados por técnicos do licenciamento e utilizados para a verificação de conformidade em modelos BIM é um dos campos promissores, principalmente a partir da utilização de linguagens visuais, que tornam os esforços de digitalização dos regulamentos mais acessíveis. Desta forma, este trabalho apresenta o desenvolvimento de um protótipo de plataforma que permite a criação de regulamentos digitais através de linguagem de programação visual baseada em blocos e a execução de verificações de conformidade a partir de modelos IFC.

1. Introdução

O licenciamento urbanístico é a ferramenta de controlo do estado para garantir o desenvolvimento de um parque edificado com características adequadas de desempenho em diversos parâmetros, p. ex. salubridade, acessibilidade, segurança e sustentabilidade [1], [2]. Em Portugal, os processos para a obtenção de uma licença ocorrem maioritariamente sob a competência das câmaras municipais, que precisam avaliar e deliberar sobre algumas das operações urbanísticas, a exemplo da construção de novos edifícios, a partir da verificação de conformidades dos projetos de obras [3]. Esta verificação parte da análise das características da edificação em função de uma série de requisitos presentes num quadro regulamentar. Contudo, este processo ainda é maioritariamente manual, recorrendo ainda, em alguns casos, a documentos impressos. Este fator contribui para a baixa eficiência, morosidade e potenciais erros para a emissão de licenças de construção [4].

A utilização do BIM para automatizar o processo de verificação de conformidades é um campo promissor, que já conta com diversos exemplos de implementação e investigação a nível nacional e internacional [4]. O caso de Singapura é um dos exemplos pioneiros de implementação, com o desenvolvimento do sistema CORENET, que permite a avaliação parcial do quadro regulamentar a partir de ficheiros IFC. Todavia, é preciso mencionar os exemplos da Finlândia [5], Noruega [6], Reino Unido e Coreia do Sul [7], que implementaram, grande parte baseados em soluções proprietárias, seus próprios sistemas de verificação. Contudo, ainda resta uma série de desafios para uma efetiva e abrangente integração do BIM a nível do licenciamento, sendo os exemplos existentes ainda limitados a algumas disciplinas, pouco flexíveis e usualmente suportados por soluções de código fechado e proprietárias [8].

Uma componente essencial de sistemas de verificação de conformidade automáticos são as regras de verificação executáveis por máquina, que são desenvolvidas através de uma interpretação da linguagem natural dos regulamentos e posterior implementação através de linguagens de programação. Esta componente também implica em grande desafio de escalabilidade nos sistemas de verificação, principalmente devido à quantidade de regulamentos que precisa ser implementado usualmente por pessoal especializado em programação.

Existem diversas estratégias para executar a interpretação de regulamentos, que partem desde a implementação direta num algoritmo [8], até a utilização de recursos de marcação semântica e machine learning [9]. Porém uma das mais estabelecidas é a RASE, que recorre a marcadores semânticos para extrair requisitos, condições de aplicabilidade e exceções dos textos em linguagem natural [10].

Já os métodos de implementação podem envolver a utilização de linguagens de programação já estabelecidas e de uso geral, como o C# [11], ou linguagens específicas de domínio para BIM [12], [13]. As linguagens específicas de domínio usualmente apoiam-se no desenvolvimento de um modelo de entidades específicas do domínio de forma a fornecer uma sintaxe mais especializada à implementação de verificações

de conformidade, contudo, estas linguagens possuem dificuldades de implementação devido à pouca quantidade de conteúdo disponível, à pequena comunidade de utilizadores e à falta de suporte. Já as linguagens de uso geral também podem implementar os modelos de entidades através de um design orientado a objetos ou a partir de ontologias [15].

Outro desafio para se implementar regulamentos através de linguagens de programação é a necessidade de pessoal técnico especializado. Desta maneira, algumas linguagens de programação visual foram desenvolvidas para tornar o processo de codificação de regulamentos mais acessível [5], [16], [17]. Estas linguagens permitem criar os regulamentos interpretáveis por máquina a partir da associação de elementos visuais como blocos e linhas de fluxo. Posteriormente, esses códigos em linguagem visual são utilizados para gerar algoritmos em linguagens tradicionais, como C# e Python, de forma a viabilizar a execução das verificações de conformidades. Os exemplos de implementação de linguagens visuais, porém, ainda são poucos e carecem demonstrar aplicações escaláveis e viáveis para a implementação.

Este trabalho tem por objetivo demonstrar a implementação de um protótipo de plataforma web para a criação de regulamentos digitais que permite codificar os requisitos do quadro regulamentar através de uma linguagem de programação visual baseada em blocos. Este sistema, por sua vez, também permite a utilização destes regulamentos digitais para a execução de verificações automáticas de conformidade a partir de modelos IFC.

O trabalho parte da análise de algumas cláusulas de regulamentos portugueses, a fim de se estabelecer um modelo de entidades do licenciamento. Seguidamente, é apresentado o desenvolvimento de linguagem de programação visual baseada no modelo desenvolvido e implementada numa plataforma web desenvolvida a partir de ferramentas e bibliotecas de código aberto. Finalmente, o desenvolvimento foi pautado em princípios de escalabilidade, com a criação de componentes que podem ser expandidos para uma maior abrangência de verificações, assim como a atualização de conceitos empregados no urbanismo.

2. Desenvolvimento de modelo de entidades do licenciamento

2.1. Análise de regulamentos e conceptualização do modelo

O desenvolvimento do modelo conceptual do licenciamento partiu da análise de 9 cláusulas objetivas do quadro regulamentar português, a fim de obter as entidades (i.e. objetos e propriedades) utilizados para a execução de verificações. Para a elaboração deste modelo, começou-se por implementar a metodologia RASE, com a marcação dos textos das cláusulas, como pode ser visto na Figura 1.

Este procedimento facilitou a obtenção das entidades das cláusulas, das condições de aplicabilidade das regras e dos requisitos a ser avaliados. Porém, apesar da sua eficiência para a avaliação de casos simples de verificação, que se baseiam na determinação do valor para propriedades de objetos, o método exige algumas adaptações para situações mais complexas. Assim, referências e exceções entre regras e regulamentos, distâncias a planos a ser verificados, entre outras situações, implicam complexidades de representação com o método que o tornam pouco eficiente.

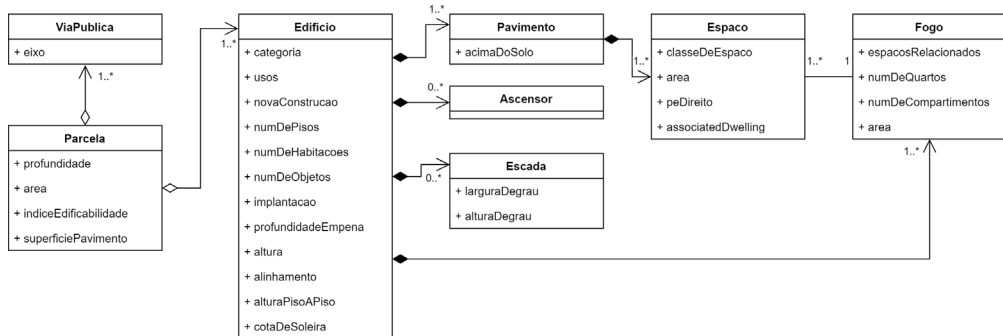
Outro ponto relevante observado na interpretação dos regulamentos foi a obtenção de definições e tratamento de ambiguidades e omissões, que impedem a correta implementação de algumas regras. Desta forma, ficou evidente que o processo de implementação dos regulamentos atuais do licenciamento é um processo a ser desenvolvido em conjunto com especialistas do urbanismo.

Figura 1
 Marcação através da metodologia RASE.

Cláusula	Texto
RGEU, Artigo 46.º 7	<p><R>Os <a>degraus das escadas das edificações para <s>habitação colectiva </s>terão a <r>largura (cobertor) mínima de 0,25 m </r> e a <r>altura (espelho) máxima de 0,193 m </r>. <E>No entanto, <s>nos edifícios de três, quatro ou cinco pisos </s> e <s>sempre que não seja instalado ascensor </s>, a <r>largura (cobertor) mínima será de 0,280m </r> e a <r>altura (espelho) máxima será de 0,175m </r>. </E></R></p>

A partir da obtenção das entidades e das suas propriedades, foi possível conceptualizar um modelo representativo do domínio do licenciamento para as cláusulas analisadas, como apresentado na Figura 2. Este modelo representa todas as entidades, propriedades e relações entre os conceitos utilizados nas cláusulas analisadas e serviu de base para a execução do mapeamento com o modelo IFC e a sua posterior implementação através da linguagem Python.

Figura 2
 Modelo conceptual de entidades desenvolvido.



2.2. Mapeamento ao modelo IFC

O procedimento central para a implementação do modelo conceptual foi o seu mapeamento com a estrutura de dados do IFC, que demonstrou ser um dos maiores desafios na implementação do sistema. Para tal, foi necessário avaliar, para cada uma das entidades do modelo conceptual, como obter a informação a partir de classes, propriedades e relações da estrutura do IFC. Todavia, esse exercício demonstrou que usualmente as relações não podem ser de 1:1, sendo que uma entidade no modelo conceptual pode necessitar ser representada através duma complexa relação entre os objetos do IFC. Para tratar destes casos mais complexos, foi preciso considerar a utilização de sistemas de classificação e processamento geométrico, que demonstraram ser essenciais para uma efetiva implementação.

O processamento geométrico foi implementado através da integração de uma biblioteca de processamento de malhas poligonais com a biblioteca de análise de ficheiros IFC, em classes que permitem a consulta ao modelo e associação de métodos de alto nível (p. ex. obtenção de volume e área de secção transversal) a cada um dos objetos obtidos. A utilização do processamento geométrico para obter informações do IFC demonstrou ser uma estratégia possível para diminuir a necessidade de criação de propriedades customizadas na modelação e consequentemente gerar requisitos de informação mais simples do ponto de vista dos dados alfanuméricos. A Tabela 1 apresenta a descrição, em alto nível de abstração, da forma pela qual algumas das propriedades dos objetos foram obtidas através de recursos de processamento geométrico.

Tabela 1: Exemplos de mapeamentos entre modelos de entidades e modelo IFC.

Parcela	
+ profundidade	A profundidade de elementos, como definida anteriormente, foi definida a partir da profundidade da 'bounding box' orientada do elemento com definição do arruamento como malha frontal.
Edifício	
+ profundidade de Implantação	A profundidade de implantação foi considerada como a distância ortogonal máxima entre o alinhamento e os elementos do edifício.
+ alinhamento	O alinhamento foi considerado como a linha formada pelos pontos mais próximos entre a projeção da parcela na maior elevação da via pública e a projeção da via pública.
Pavimento	
+ areaBruta	A área bruta do pavimento foi considerada como a área de todos os espaços de um pavimento, mais as áreas das projeções das paredes internas e externas ao nível do piso, sem considerar as áreas das varandas.
Espaco	
+ area	A área do espaço foi considerada como a área da secção transversal do espaço considerado como fogo, na altura do plano definido pelo seu centroide.

A utilização de um sistema de classificação demonstrou ser uma boa estratégia para casos como obtenção de compartimentos duma edificação, p. ex. cozinha, varanda, sala. O sistema escolhido foi o SECCLasS, já que o seu desenvolvimento parte da

implementação de particularidades do setor da construção português. Contudo, verificou-se que em alguns casos o sistema é incompleto ou pouco especializado para o domínio do licenciamento, o que implicou na criação de novos códigos para a representação de entidades como “quartos duplos” ou “arrecadações”. A classificação também demonstrou ser uma alternativa para determinar elementos como o “fogo”, “parcela” e a “cota de soleira”.

Parte do desafio do mapeamento também partiu das limitações de modelação presentes nos softwares proprietários, como por exemplo a impossibilidade de criar espaços anexados a outros espaços, o que é previsto no modelo IFC a partir do atributo “CompositionType”.

Na Tabela 2, é possível observar o mapeamento de algumas das entidades e propriedades que não necessitaram de processamento geométrico e recorreram ao uso de classificação, propriedades nativas do IFC (NP) e propriedades customizadas (PP).

Tabela 2: Exemplos de mapeamentos entre modelos de entidades e modelo IFC.

Modelo de entidades	Modelo IFC
Parcela	IfcSpace com classe “SL_22_05”
ViaPublica	IfcSpace com classe “SL_22_10”
Edifício	IfcBuilding
+ categoria	PP: Category
+ usos	PP: Uses
+ novaConstrução	PP: IsNewConstruction
+ cotaSoleira	Propriedade Elevation do IfcBuildingStorey com classe “En_95_05”
Elevador	IfcTransportElement.ELEVATOR
Escada	IfcStair
+ larguraDegrau	NP: TreadLength
+ alturaDegrau	NP: RiserHeight
Pavimento	IfcBuildingStorey
Fogo	IfcSpace com classe “SL_22_15”
+ referência	PP: DwellingReference
Espaco	IfcSpace
+ classeDeEspaco	Classificação (IfcClassification)
+ fogoAssociado	PP: AssociatedDwelling

3. Criação de linguagem visual e implementação de protótipo de plataforma

3.1. Linguagem de programação visual

A criação da linguagem visual foi assistida pela biblioteca de código aberto Blockly, baseada em JavaScript, e desenvolvida pela google. Esta biblioteca permite a criação duma linguagem de programação baseada em blocos que gera outra linguagem baseada em script a ser interpretada. Desta forma, o desenvolvimento da linguagem foi

executado a partir de duas atividades principais: (i) criação da sintaxe da linguagem (ii) desenvolvimento dos geradores de código.

A abordagem adotada na primeira tarefa consistiu em criar diferentes categorias de blocos para representar os elementos necessários para a criação de regras de verificação. Assim, foram criadas 5 categorias de código, demonstrados na Figura 3: (a) blocos representativos do modelo de entidades e suas propriedades; (b) blocos que representam declarações e operadores lógicos; (c) blocos que determinam os requisitos que devem ser verificados; (d) blocos para representar operações matemáticas e (e) blocos que representam valores.

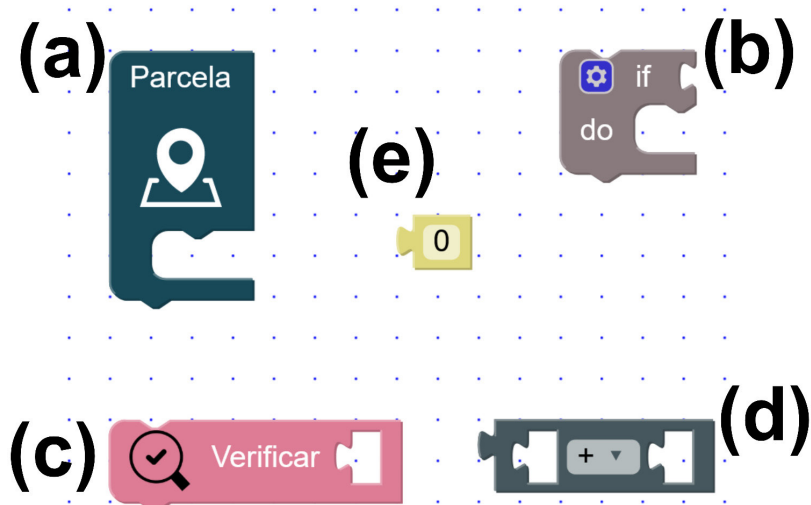


Figura 3
Categorias de blocos desenvolvida: (a) blocos do modelo; (b) blocos de lógica; (c) blocos de verificação; (d) blocos de matemática; (e) blocos de valores.

As categorias desenvolvidas possuem blocos capazes de executar todas as verificações analisadas, como representado na Figura 4 e potencialmente executar diversas outras verificações a partir da criação de outros algoritmos e alteração de valores. Outra vantagem da categorização foi a implementação duma identidade visual, o que facilita o entendimento e utilização da linguagem. Além disso, essas categorias podem servir de base para a expansão através de novos blocos e conseqüentemente uma capacidade mais abrangente de codificação de regras.

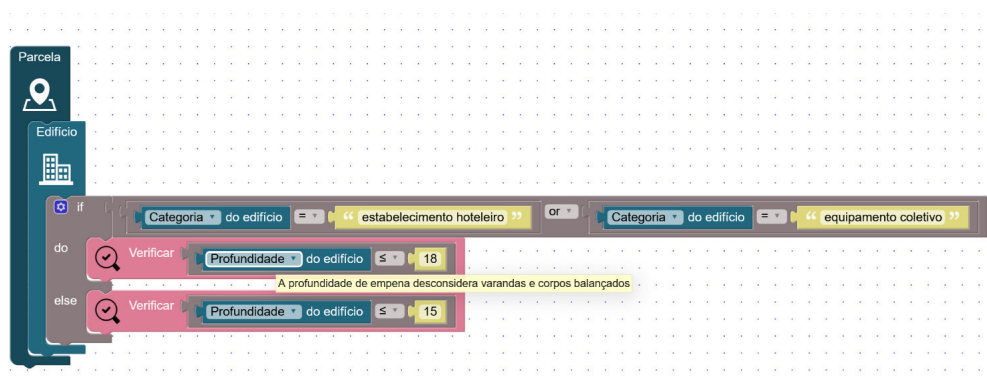


Figura 4
Regra em blocos para o RPDML Artigo 43.º, parágrafo 1.

3.2. Implementação de protótipo de plataforma web

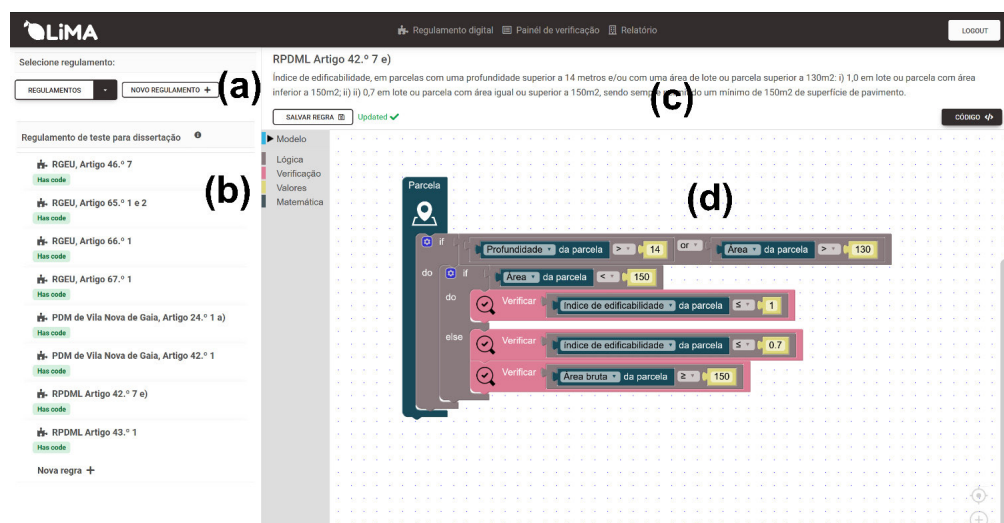
O desenvolvimento do protótipo de plataforma web foi largamente suportado pela utilização de *frameworks* de código aberto, que permitiram criar uma plataforma dinâmica, capaz de integrar os componentes de visualização de modelos IFC e criação de regulamentos digitais de forma modular. Assim sendo, o desenvolvimento da plataforma pôde ser dividido em duas principais tarefas: (i) desenvolvimento dos serviços a nível do servidor; (ii) desenvolvimento dos serviços a nível do navegador web.

Ao nível do servidor ocorreu o desenvolvimento da base de dados, da criação de métodos para executar verificações e da API para permitir o fluxo de informações com os serviços executados no navegador. Neste nível ocorre o armazenamento dos regulamentos interpretáveis por máquina e a sua execução, por métodos apropriados que avaliam as regras criadas em linguagem Python que remetem ao modelo de entidades. Desta forma, foi crucial desenvolver uma base de dados que estivesse adequada à representação dos regulamentos e às suas regras. Além disso, a API desenvolvida permite transferir os regulamentos entre o servidor e o navegador através do formato JSON, que integra os códigos Python representativos das regras, quando necessário.

Ao nível do navegador, foram desenvolvidos componentes para a criação e edição dos regulamentos, assim como para a execução de verificações e visualização de resultados. A criação de regulamentos envolve a inserção de informações básicas sobre o regulamento que se pretende desenvolver e a posterior criação e codificação de regras com os blocos de programação visual. Também foi necessário permitir a associação a referências e os textos originais das cláusulas dos regulamentos. A página para a criação e edição de regulamentos pode ser visualizada na Figura 5. Como é possível observar, a codificação dos regulamentos digitais tornar-se bastante simples.

Figura 5

Página de criação de regulamentos: (a) seletor de regulamentos; (b) lista de regras; (c) texto original da cláusula; (d) editor de regras de verificação.



Já a verificação envolve o carregamento do ficheiro IFC que se pretende avaliar e a seleção dos regulamentos que se pretende verificar. Posteriormente, é possível enviar a requisição para execução da verificação, que remete as informações do ficheiro e avalia consoante as regras codificadas dos regulamentos, apresentando uma série de resultados para cada uma das regras avaliadas. Houve também a tentativa de se desenvolver métodos de visualização (Figura 6) que permitissem verificar o modelo consoante o resultado da verificação, que apesar de ter potencial, evidenciou que é necessário estabelecer métodos que tenham representações adequadas ao tipo de verificação executado.

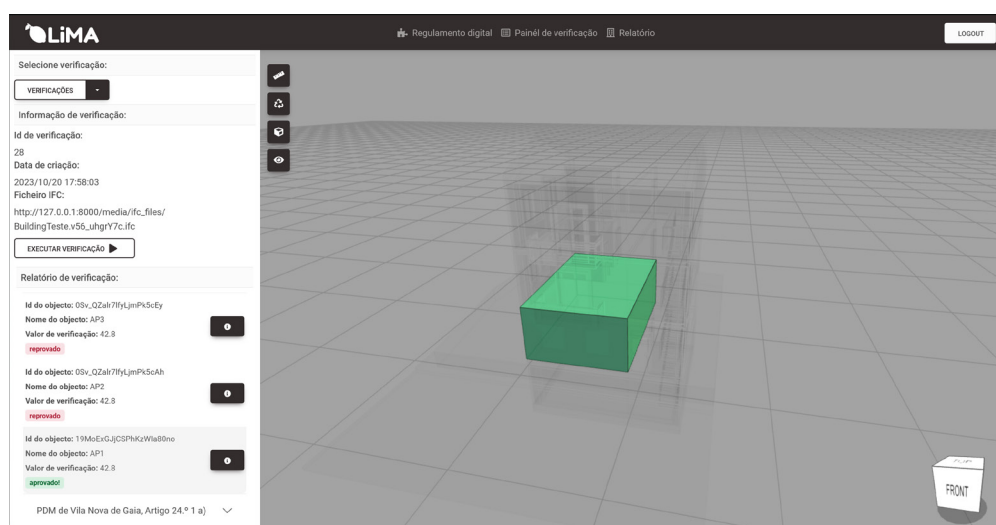


Figura 6
Relatório de verificação
assistido por
visualizador.

4. Conclusão

Neste trabalho é apresentado um protótipo de plataforma web para a criação de regulamentos digitais que permite codificar os requisitos do quadro regulamentar através de uma linguagem de programação visual baseada em blocos. Através da implementação e análise de alguns regulamentos foi possível verificar a viabilidade e avaliar diversos desafios da implementação de sistemas para a criação de regulamentos digitais e de sistemas para a automatização da verificação de conformidades no âmbito do licenciamento urbanístico.

Foi possível concluir que é exequível automatizar parcialmente a verificação de conformidades das operações urbanísticas em função dos requisitos presentes no quadro regulamentar. Além disso, a utilização de um modelo de entidades que permita trabalhar com conceitos do licenciamento ao invés de trabalhar com o modelo IFC diretamente é uma boa estratégia. Neste caso, o modelo foi implementado através duma série de classes em Python, contudo outros recursos podem ser utilizados, tais como o recurso a ontologias do ambiente construído.

Devido às subjetividades, ambiguidades e omissões das cláusulas dos regulamentos, a interpretação dos regulamentos é uma atividade que requer a consulta a especialistas do licenciamento para a correta obtenção das entidades.

A utilização de processamento de informação geométrica do modelo IFC para alimentar o modelo conceptual é uma estratégia que permite resolver alguns problemas de entidades complexas do ponto de vista geométrico e possibilita a redução dos requisitos de modelação. No entanto, outras estratégias devem ser utilizadas em conjunto, como a obtenção de propriedades de objetos e sistemas de classificação.

A linguagem de programação visual permite a criação e edição de regulamentos digitais de forma acessível, o que facilita a utilização por técnicos do licenciamento e tornam a solução muito mais escalável.

Agradecimentos

Este trabalho foi parcialmente financiado pela FCT/MCTES através de fundos nacionais (PIDDAC) no âmbito da unidade de P & D do Instituto para Sustentabilidade e Inovação em Engenharia Estrutural (ISISE), com referência UIDB / 04029/2020 (doi.org/10.54499/UIDB/04029/2020), e no âmbito do Laboratório Associado de Produção Avançada e Sistemas Inteligentes ARISE com a referência LA/P/0112/2020. Este trabalho foi também financiado pela bolsa de doutoramento (BI-ISISE-UMINHO-CHEK-B03) atribuída ao primeiro autor, financiada pelo Programa de Investigação e Inovação Horizonte Europa 2021-2027, no âmbito do projeto Change Toolkit for Digital Building Permit CHEK, com referência 101058559. Agradeço também aos demais parceiros do projeto CHEK, ao nível da Câmara Municipal de Gaia (Carla Malta Pires, Ana Patrícia Baptista e Marco Lima de Carvalho), da Câmara Municipal de Lisboa (André Espinho e João Frescata Pereira) e da Universidade Lusófona (Ricardo Mateus).

Referências

- [1] J. B. Pedro, F. Meijer, e H. Visscher, «Comparison of building permit procedures in European in European Union countries Cova da Moura Rehabilitation Needs Assessment View project Building Regulations for Construction Works in Existing Buildings View project Comparison of building permit procedures in European Union countries», 2011. [Em linha]. Disponível em: <https://www.researchgate.net/publication/257527312>
- [2] J. Branco Pedro, F. M. Meijer, e H. J. Visscher, «The Portuguese Building Regulation System: A Critical Review», 2009, Acedido: 7 de agosto de 2023. [Em linha]. Disponível em: <http://repositorio.lnec.pt:8080/xmlui/handle/123456789/16726>
- [3] «Regime jurídico da urbanização e edificação – RJUE | DRE». Acedido: 29 de maio de 2023. [Em linha]. Disponível em: <https://dre.pt/dre/legislacao-consolidada/decreto-lei/1999-34567875>

- [4] F. Noardo *et al.*, «Unveiling the actual progress of Digital Building Permit: getting awareness through a critical state of the art review», 2021. [Em linha]. Disponível em: <https://www.bimacademy.global/insights/infrastructure/the-golden-thread-of-information-putting-the-hacki>
- [5] C. Preidel e A. Borrmann, «Automated code compliance checking based on a visual language and building information modeling», *32nd Int. Symp. Autom. Robot. Constr. Min. Connect. Future Proc.*, 2015, doi: 10.22260/ISARC2015/0033.
- [6] E. Hjelseth, «Public BIM-based model checking solutions: lessons learned from Singapore and Norway», em *Building Information Modelling (BIM) in Design, Construction and Operations*, WIT Press, set. 2015, pp. 421-436. doi: 10.2495/bim150351.
- [7] I. Kim, J. Choi, E. A. L. Teo, e H. Sun, «Development of kbim e-submission prototypical system for the openbim-based building permit framework», *J. Civ. Eng. Manag.*, vol. 26, n.º 8, pp. 744-756, nov. 2020, doi: 10.3846/jcem.2020.13756.
- [8] N. O. Nawari, «A Generalized Adaptive Framework (GAF) for automating code compliance checking», *Buildings*, vol. 9, n.º 4, abr. 2019, doi: 10.3390/buildings9040086.
- [9] J. Zhang e N. M. El-Gohary, «Extending Building Information Models Semiautomatically Using Semantic Natural Language Processing Techniques», *J. Comput. Civ. Eng.*, vol. 30, n.º 5, p. C4016004, set. 2016, doi: 10.1061/(ASCE)CP.1943-5487.0000536.
- [10] E. Hjelseth e N. N. Nisbet, «Capturing normative constraints by use of the semantic mark-up RASE methodology», 2011. Acedido: 22 de fevereiro de 2023. [Em linha]. Disponível em: <https://www.semanticscholar.org/paper/Capturing-normative-constraints-by-use-of-the-RASE-Hjelseth-Nisbet/8820d782f37a3bccc5922dc5e6ee1b344a027828>
- [11] M. F. de S. Santos, «Metodologias BIM para verificação regulamentar em contexto de licenciamento municipal: proposta, implementação e aplicação», masterThesis, 2021. Acedido: 14 de agosto de 2023. [Em linha]. Disponível em: <https://repositorium.sdum.uminho.pt/handle/1822/76900>
- [12] J. K. Lee, «Building environment rule and analysis (BERA) language and its application for evaluating building circulation and spatial program», Georgia Institute of Technology, 2011. Acedido: 3 de março de 2023. [Em linha]. Disponível em: <https://smartech.gatech.edu/handle/1853/39482>
- [13] J. Dimyadi, «Integrating the BIM Rule Language into Compliant Design Audit Processes», 2016.

- [14] A. Yurchyshyna e A. Zarli, «An ontology-based approach for formalisation and semantic organisation of conformance requirements in construction», *Autom. Constr.*, vol. 18, n.º 8, pp. 1084-1098, dez. 2009, doi: 10.1016/j.autcon.2009.07.008.
- [15] H. Kim, J. K. Lee, J. Shin, e J. Choi, «Visual language approach to representing KBimCode-based Korea building code sentences for automated rule checking», *J. Comput. Des. Eng.*, vol. 6, n.º 2, pp. 143-148, abr. 2019, doi: 10.1016/J.JCDE.2018.08.002.
- [16] A. Wülfing, R. Windisch, e R. Scherer, «A visual BIM query language», em *eWork and eBusiness in Architecture, Engineering and Construction*, A. Mahdavi, B. Martens, e R. Scherer, Eds., CRC Press, ago. 2014, pp. 157-164. doi: 10.1201/b17396-30.