Realidade Aumentada para a detecção de conflitos em projetos: proposta de aplicativo

Augmented Reality for clash detection in design: application proposal

Cauê Carneiro Santiago
Universidade Estadual de Campinas | Campinas | Brasil | 226650@dac.unicamp.br
Danielle Skubs
Universidade Estadual de Campinas | Campinas | Brasil | 002902@dac.unicamp.br
Ana Regina Mizrahy Cuperschmid
Universidade Estadual de Campinas | Campinas | Brasil | cuper@unicamp.br

Resumo

Sistemas de compatibilização de projetos oferecem recursos para detecção de conflitos, ajudando no processo de revisão das disciplinas envolvidas. Por outro lado, a imersão e interação dos projetistas em tempo e escala reais, proporcionada pela Realidade Aumentada (RA), favorecem o processo de projeto na etapa de concepção, principalmente, nas tomadas de decisão e nos processos de colaboração. Tendo em vista esses benefícios, o objetivo deste artigo é propor uma alternativa para detecção de colisão de projetos, oferecendo a visualização dos componentes conflitantes em RA. Essa foi uma pesquisa exploratória que indicou possibilidade de aplicação de RA como apoio nesse processo.

Palavras-chave: Realidade Aumentada. Realidade Misturada. Compatibilização de Projetos. Detecção de Colisão. Detecção de Conflitos.

Abstract

Design coordination systems offer resources for clash detection, helping in the design review process. However, the designers' immersion and interaction in real time and scale, provided by Augmented Reality (AR), favor the design process in the conceptual design stage, mainly in decision making and collaboration processes. In view of these benefits, this article proposes an alternative for clash detection, offering AR visualization of spatial conflicts. This was an exploratory research that indicated the possibility of applying AR as a support in this process.

Keywords: Augmented Reality. Mixed Reality. Project Compatibility. Clash Detection. Conflict Detection.

Como citar:
SANTIAGO, C. C. .; SKUBS , D. .; CUPERSCHMID , A. R. M. Realidade Aumentada para a detecção de conflitos em projetos: proposta de aplicativo. In: SIMPÓSIO BRASILEIRO DE TECNOLOGIA DA INFORMAÇÃO E COMUNICAÇÃO NA CONSTRUÇÃO, 3., 2021, Uberlândia. **Anais** [...]. Porto Alegre: ANTAC, 2021. p. 1-10. Disponível em: https://eventos.antac.org.br/index.php/sbtic/article/view/581. Acesso em: 3 ago. 2021.

INTRODUÇÃO

A realidade misturada, junção em diferentes proporções do campo real com as realidades virtual e o mundo físico, tem sido utilizada já há algum tempo em diversas áreas e auxilia na visualização e compreensão de conceitos [1].

Na área da Arquitetura, Engenharia e Construção (AEC), vale ressaltar que as experiências em Realidade Aumentada (RA) facilitam a visualização rápida e em tempo real em diferentes ângulos e escalas de um projeto ou modelo em campo e, em contrapartida, as experiências com foco em Realidade Virtual (RV), melhoram o processo de imersão do usuário, geralmente através de dispositivos do tipo *Head Mounted Display* (HMD) [2].

A comercialização em massa de dispositivos portáteis como o *smartphones* e o desenvolvimento de *software* e aplicativos têm tornado essas tecnologias mais acessíveis. Apesar disso, o desenvolvimento em projetos de Arquitetura com o auxílio de RV e RA não vem acontecendo com a rapidez esperada devido a questões técnicas e custos de dispositivos específicos que atendem a estes tipos de imersão em outras realidades [3].

Ainda assim, a comercialização de dispositivos para RV, como Samsung Gear VR ou até mesmo o Google CardBoard, que funcionam como *headsets* dependentes de um *smartphone* para processamento e visualização das informações, tem tornando o acesso a experiências em RV mais viável para universidades, faculdades e escritórios de projetos [4].

Já em RA, tem-se visto, de forma mais ampla, o desenvolvimento de aplicações para uso em *smartphones e tablets*, facilitando o acesso a experiências que auxiliem nos processos de projeto [3]. O autor cita ainda que a produção de projetos diretamente em ambiente virtual é mais difundida pela indústria mecânica, como nos processos de criação de design de automóveis por exemplo, do que em AEC e que as plataformas de imersão em RV para design são mais utilizadas nas etapas iniciais de projeto, contribuindo no processo criativo, contextualização do design, flexibilidade nas escalas do modelo e representação explícita do mesmo.

A imersão e interação dos projetistas em tempo e escala real, *in loco*, proporcionada pela RA, favorecem o processo de projeto na etapa de concepção, principalmente, nas tomadas de decisão e nos processos de colaboração em AEC [5].

Essas características podem ser aproveitadas também para estudos de compatibilização de projetos das diversas disciplinas envolvidas em AEC. Atualmente, o processo de compatibilização de projetos em 3D é realizado com apoio de ferramentas *Building Information Modeling* (BIM) disponíveis comercialmente. Entretanto, tais ferramentas não possibilitam a visualização em RA. Parte-se do pressuposto que a tecnologia de RA poderia propiciar uma alternativa para visualização de conflitos em projetos.

A visualização das interferências por meio de RA foi demonstrada em pesquisa anterior [6], porém, a falta de destaque dos componentes que colidiram dificultava a

identificação dos mesmos no modelo. Considerando essa questão, esta pesquisa apresenta uma prova de conceito que evidencia a viabilidade do processo de detecção de conflitos através da visualização em RA, destacando graficamente os elementos colididos. Assim, o objetivo deste artigo é **propor uma alternativa para detecção de colisão de projetos, oferecendo a visualização dos componentes conflitantes em RA.**

A DETECÇÃO DE CONFLITOS EM PROJETOS

A detecção prévia de conflitos em projetos já é possível através de *software* específicos e auxiliam não só os projetistas, mas todos os envolvidos a repensar possíveis falhas de projeto antes de sua implantação ou execução [7]. A detecção de conflitos em projetos através de sistemas BIM possibilita maior controle e precisão durante a análise, uma vez que o sistema oferece maior quantidade de informações e detalhes projetuais, desde que devidamente detalhadas e pensadas durante o desenvolvimento do projeto em qualquer plataforma BIM [8].

Ferramentas como o Navisworks [9], Solibri [10] e Trimble Connect [11] oferecem recursos para rápida e precisa detecção de conflitos em projetos, ajudando os projetistas e envolvidos a revisar e estudar os problemas entre as diversas disciplinas de projetos. Todas essas incompatibilidades se traduzem em custos, muita das vezes, elevados, a depender de sua gravidade [12].

Porém, da mesma forma que é possível detectar com precisão conflitos em projetos, é preciso ter cautela com o objetivo da análise uma vez que projetos de arquitetura e estruturas, por exemplo, podem apresentar conflitos não expressivos para compatibilização, como lajes engastadas em vigas, pilares conectados a paredes de vedação ou até mesmo tubulação hidráulica embutida em paredes [13]. Assim, é preciso analisar e definir a tolerância da detecção de colisão de elementos ou até mesmo a classe de elementos sobrepostos que o *software* deve analisar [9]. Neste sentido, faz-se necessário categorizar elementos e sistemas, possibilitando a escolha seletiva para a detecção de conflitos desejada.

MATERIAS E MÉTODOS

O presente estudo busca, através de estudo exploratório e de caráter qualitativo, propor uma alternativa, em RA, para a detecção de conflitos em projetos. Para tanto, envolve o desenvolvimento de um aplicativo específico e avalia seu desempenho.

O aplicativo em RA foi desenvolvido em plataforma Android, e checa a detecção de colisão entre objetos dentro de um modelo tridimensional proveniente de *software* BIM. Esse modelo, foi pré-inserido na *engine* Unity e, por meio de programação na linguagem específica C#, possibilita a detecção das colisões e as indica através de diferentes cores dos objetos colididos, além de informações textuais ou ícones que representam visualmente os pontos de colisão. Posteriormente, foi avaliado o suporte desta estratégia para o processo de compatibilização de projetos.

Para o experimento foi gerado um modelo tridimensional de projeto estrutural desenvolvido e extraído do *software* TQS em arquivo de extensão IFC - o que garante

uma maior integridade das informações de cada objeto representado no modelo. Tendo em vista que o Unity não aceita, de forma nativa, arquivos em IFC, o mesmo teve que ser importado para o Revit para ser, posteriormente, exportado em FBX - compatível com o Unity.

O *software* utilizado para desenvolvimento da aplicação foi o Unity (versão 2019.3.0.a2), com programação em C# e Software Development Kit (SDK) VUFORIA - que oferece suporte para aplicações em RA. A partir disso, foi possível gerar um aplicativo em RA para *smartphone* com plataforma Android. Posteriormente, o desempenho do aplicativo foi avaliado (Figura 1).

Figura 1: Metodologia de desenvolvimento do aplicativo em RA a partir de modelo 3D em BIM



Fonte: os autores.

Para avaliar o aplicativo, foi utilizado um *smartphone Samsung Galaxy S7 Edge*, que possuía todos os aparatos necessários para que o aplicativo funcionasse, como o acelerômetro e o giroscópio.

DESENVOLVIMENTO DO APLICATIVO

Para contextualizar o desenvolvimento do aplicativo, foi utilizado um modelo estrutural para avaliar a viabilidade da detecção de colisões entre seus componentes através da visualização destas em RA.

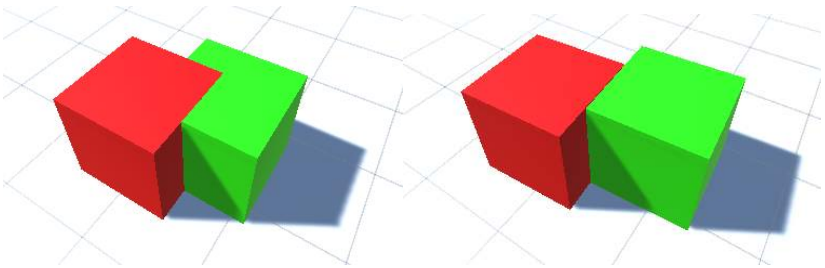
Inicialmente, foi necessário verificar o suporte da *engine* Unity tendo em vista as possibilidades de interação física dos objetos. Para isso, foram experimentadas diversas funções que poderiam dar suporte à checagem de colisões.

Em uma primeira experimentação, foram criados inicialmente dois cubos e um plano, chamados também de *Game Objects* no Unity. Apenas aos cubos, foi atribuído um componente de corpo rígido (*Rigid Body)* e atribuída a propriedade de gravidade, inerente ao componente. Essa *engine* trabalha com um sistema de coordenadas tridimensionais, em “x”, “y” e “z”. Os cubos foram espaçados com certa distância em

um dos eixos e foi aplicado a um deles um *script* em C# para que o mesmo se movimentasse. Ao término de sua movimentação, os cubos deveriam colidir. O experimento permitiu, mesmo que de forma simples, a observação do comportamento da física no Unity.

Na sequência, outra experimentação foi realizada. Nela, os cubos foram posicionados um sobreposto ao outro (Figura 02 - esquerda) e o *script* de movimentação foi removido. Essa configuração buscava criar uma situação para analisar modelos tridimensionais de elementos estáticos - com o intuito de verificar as colisões entre elementos construtivos. Observou-se então que os cubos, por estarem com os componentes de corpo rígido e gravidade aplicados, se repeliram (Figura 02 - direita), indicando a necessidade de ancoragem dos objetos.

Figura 2: Sobreposição dos cubos inicialmente e após, ao se repelirem



Fonte: os autores.

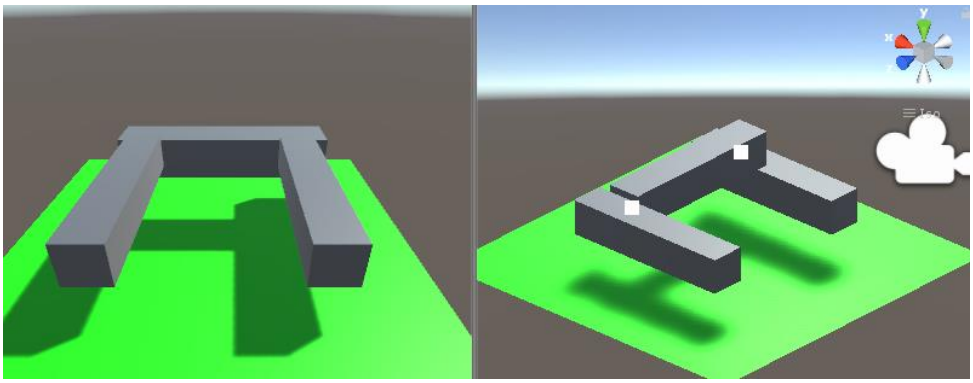
Nas opções de corpo rígido foi possível obter o travamento dos elementos ao se congelar as posições e rotações nos três eixos (x, y, z). Essas variáveis foram aplicadas a ambos os cubos, resolvendo o problema apresentado.

Ademais, fez-se necessário para o experimento, que o Unity indicasse visualmente a colisão. Para isso, foi criado um script em C# que aponta a colisão. Percebeu-se que o Unity é capaz de apontar a colisão, mas somente em forma de *debug* ou depuração interna, o que não seria uma estratégia adequada, por não ser visual. Partiu-se então para o desenvolvimento de uma função que indicaria visualmente o ponto de colisão na cena e não somente no depurador interno.

Para tal, foi utilizada a função *Raycast* nativa do Unity, que indica, por meio de linhas, os pontos de colisão entre objetos. Ainda assim, a função também não se mostrou a melhor solução, pois criou uma linha do centro de coordenadas (ponto 0,0,0) até os pontos de interpolação dos objetos, não mostrando de uma forma clara a colisão em si.

Outra experimentação envolveu a criação de *Gizmos*, sobreposições gráficas, que oferecem depuração visual. Essas sobreposições podem tomar formas de cubos, esferas, entre outros. Os *Gizmos* funcionaram como esperado, pois, indicaram exatamente os pontos de colisão, facilitando a verificação e deixando a informação mais clara. No entanto, esse é um processo interno do Unity, que não é exibido na execução do aplicativo para o usuário final - o que inviabilizou este recurso (Figura 03).

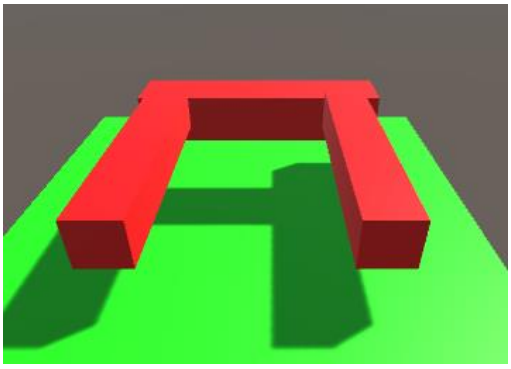
Figura 3 - Modelo com Gizmos: cena iniciada (esquerda) e depuração interna (direita)



Fonte: os autores.

Com isso, foi adotada uma nova estratégia que visava a alteração da cor dos elementos colididos, através de um *script* em C# (Figura 4). Essa solução se mostrou adequada para verificação de colisão entre os elementos, evidenciando a colisão encontrada.

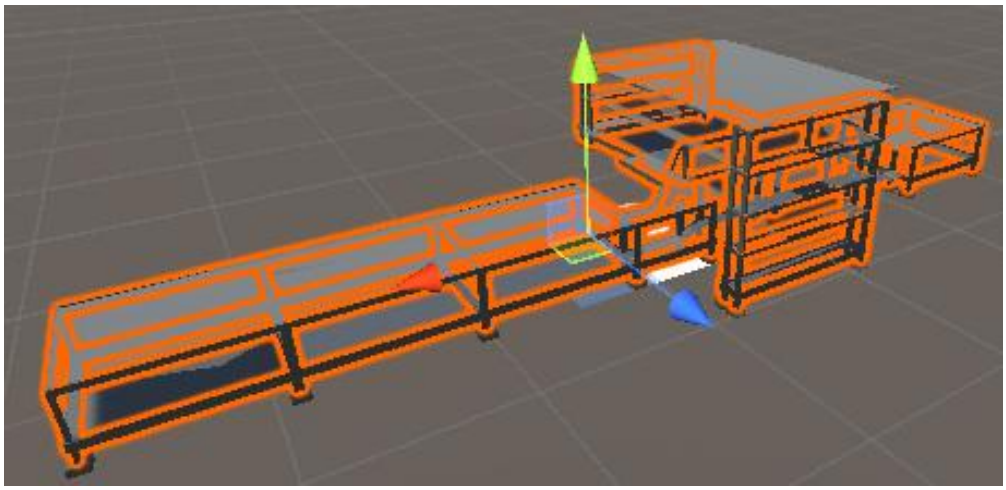
Figura 4 - Modelo com ajuste de cor na colisão



Fonte: os autores.

Na sequência, foi inserido o modelo estrutural de uma edificação. Nas primeiras experimentações foi utilizado um modelo em formato FBX, que ao ser importado no Unity, criou, por padrão, um agrupamento de elementos. Nessa situação, todos os pilares, as vigas e as lajes do modelo foram condensados em uma única instância, inviabilizando a verificação de colisão dos componentes (Figura 5). Isso porque, qualquer elemento, ao colidir com qualquer parte do modelo (formado por elementos condensados), causava alteração de cor em todos os componentes, uma vez que todos estavam em colisão por compartilharem alguma face ou aresta.

Figura 5 - Modelo complexo agrupado nativamente



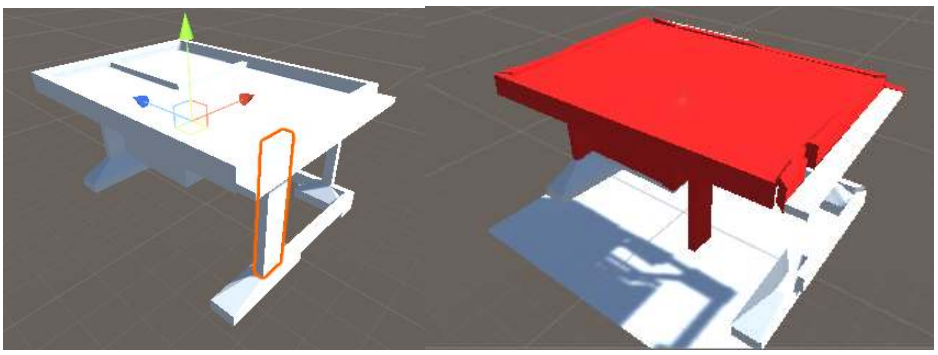
Fonte: os autores.

Posteriormente, foi verificada outra forma de exportação do modelo em IFC a partir do *software* TQS. Desta vez, foi utilizada a opção de configuração otimizada para o Revit 2011 ou superior. No Revit, o modelo foi exportado para FBX para ser importado para o Unity. Nesse caso, observou-se que todos os elementos da estrutura não foram agrupados, viabilizando a verificação de colisões entre os componentes.

A partir disso, para cada um dos objetos foram estabelecidas a configurações de corpo rígido e colisor de malha, além de vínculo com o *script* que ajusta a cor dos objetos colididos. Em um primeiro momento, não foi detectada nenhuma colisão, pois o modelo não apresentava conflitos.

Para validar a mudança de cor mediante a colisão, foi realizada uma alteração na posição de um dos pilares do modelo, garantindo que houvesse a colisão. Assim, a colisão foi detectada e ambos os componentes que estavam colidindo tiveram as cores ajustadas como esperado (Figura 6).

Figura 6 - Parte da estrutura com componentes desagrupados e indicação de elementos colididos



Fonte: os autores.

Para desenvolver a aplicação em RA, foi utilizada a *engine* Vuforia [14]. Foi gerado um aplicativo para *smartphones* da plataforma Android, que utiliza como marcador o VuMark (Figura 7), responsável por conectar o modelo virtual com o ambiente real.

Figura 7: Marcador utilizado: VuMark



Fonte: Biblioteca Vuforia (2019).

Para finalizar, foi desenvolvida a interface com o usuário. Para tanto, foram feitos ajustes na resolução e no *canvas*, de forma a se adequar à tela de um *smartphone*. Nela, foi criado um botão para que durante o uso do aplicativo, o *script* que detecta as colisões fosse iniciado.

AVALIAÇÃO DE DESEMPENHO DO APLICATIVO

O aplicativo desenvolvido foi avaliado tendo em vista seu desempenho em exibir as colisões encontradas e permitir sua visualização em RA (Figura 8). Seu funcionamento se mostrou apropriado, respondendo ao comando do usuário de verificar conflitos e permitindo a visualização de diversos ângulos em RA. O modelo permaneceu estável, fixo em um dado local, graças à utilização de um marcador.

O aplicativo apresentou de forma visual a colisão encontrada, destacando em vermelho os elementos colididos. A informação apresentada, facilitou a identificação de tais elementos.

Figura 8: Modelo visto de ângulos diferentes a partir do aplicativo, com destaque para os conflitos encontrados



Fonte: os autores.

Comparativamente ao uso de *software* específico para detecção de conflitos [9], nesse aplicativo, a tolerância da detecção de colisão entre os elementos foi pré-estabelecida por programação, e não por uma interface com o usuário que permite alteração nos valores. Adicionalmente, não foi possível categorizar elementos e sistemas para o processo, limitando sua aplicação para avaliação entre disciplinas diferentes. Isso se justifica por este estudo ser uma prova de conceito das alternativas de visualização dos conflitos.

CONCLUSÕES

Este artigo apresentou, através do desenvolvimento de um aplicativo, uma prova de conceito para viabilidade de desenvolvimento de um sistema que apoiasse o processo de detecção de conflitos e propiciasse a visualização em RA. A contribuição desta pesquisa reside em validar a viabilidade desse sistema. Nele, os elementos colididos são destacados em cores diferentes, além de permitir a observação em RA. Esse é um importante avanço em relação à pesquisa anterior [6].

Entretanto, cabe destacar que esse foi um processo exploratório que exigiu um processo manual envolvendo a conversão do modelo BIM para um formato determinado, programação em C# e uso de *software* não comum à área de AEC, como o Unity. Esses fatores limitam o desenvolvimento de aplicações semelhantes, pois exigem que sejam realizadas customizações que demandam conhecimentos específicos. Ademais, o processo de desenvolvimento de aplicativos é moroso e complexo.

Trabalhos futuros poderiam investigar as possibilidades de incorporação da tecnologia de RA em *software* comercial próprio para detecção automática de conflitos, de forma a facilitar a utilização por profissionais de AEC. Para tanto, seria necessário investigar as *Application Programming Interfaces* (APIs) disponibilizadas pelos fabricantes para averiguar o suporte à implementação de RA.

AGRADECIMENTOS

Sinceros agradecimentos ao suporte dado pela Faculdade de Engenharia Civil, Arquitetura e Urbanismo da Universidade Estadual de Campinas (UNICAMP).

REFERÊNCIAS

[1] SCHNABEL, M. A.; WANG X. **Mixed Reality in Architecture, Design & Construction.** New York, NY: Springer-Verlag New York Inc. 2009.

[2] MESSADI, T. *et al*. Immersive Learning for Sustainable Building Design and Construction Practices. **Journal of Civil Engineering Architecture,** n.11, p. 841-852, 2017. DOI: 10.17265/1934-7359/2017.09.003

[3] MILOVANOVIC, J. *et al.* Virtual and Augmented Reality in Architectural Design and Education: An Immersive Multimodal Platform to Support Architectural Pedagogy In: INTERNATIONAL CONFERENCE ON COMPUTER AIDED ARCHITECTURAL DESIGN FUTURES, 17., 2017, Istanbul, Turkey. **Proceedings [...].** Instalbul. CAADFutures, p. 513-532, 2017.

[4] GOOGLE. **Google Cardboard.** Disponível em: https://arvr.google.com/cardboard/. Acesso em: 09 fev. 2021.

[5] CUPERSCHMID, A. R. M. **Realidade Aumentada no processo de projeto participativo arquitetônico:** desenvolvimento de sistema e diretrizes para utilização. Tese (Doutorado em Arquitetura e Urbanismo) - Faculdade de Engenharia Civil, Arquitetura e Urbanismo, Universidade Estadual de Campinas. Campinas, 315 p., 2014.

[6] OLIVEIRA, R. R. de .; CUPERSCHMID, A. R. M. . BIM associado à Realidade Aumentada no processo de compatibilização de projetos. In: SIMPÓSIO BRASILEIRO DE TECNOLOGIA DA INFORMAÇÃO E COMUNICAÇÃO NA CONSTRUÇÃO, 2., 2019. **Anais [...].** Porto Alegre:

ANTAC, 2019. p. 1–8. Disponível em: https://eventos.antac.org.br/index.php/sbtic/article/view/145. Acesso em: 31 jul. 2021.

[7] DANIOTTI, B.; GIANINETTO, M.; DELLA TORRE, S. (Eds.). Digital Transformation of the Design, Construction and Management Processes of the Built Environment. **Cham: Springer International Publishing**, 2020. p. 115–125. DOI: https://doi.org/10.1007/978-3-030-33570-0

[8] EASTMAN, C. *et al.* **BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors**. 2. ed. New Jersey: Wiley Publishing, 2011.

[9] AUTODESK. **Navisworks**. Disponível em: https://www.autodesk.com/products/navisworks/features. Acesso em: 09 fev. 2021.

[10] NEMETSCHEK. **Solibri**. Disponível em: https://www.solibri.com/our-offering. Acesso em: 09 fev. 2021.

[11] TRIMBLE. **Trimble Connect**. Disponível em: https://connect.trimble.com/capabilities. Acesso em: 09 fev. 2021.

[12] PEREIRA, L. M. *et al.* Estudo exploratório comparativo da eficácia entre protótipos físico, analítico 2D e 3D na identificação de inconsistências de projetos. **Gestão e Tecnologia de Projetos**, v. 10, n. 1, p. 29–47, 2015. DOI: https://doi.org/10.11606/gtp.v10i1.83373

[13] PÄRN, E. A.; EDWARDS, D. J.; SING, M. C. P. Origins and probabilities of MEP and structural design clashes within a federated BIM model. **Automation in Construction**, v. 85, p. 209–219, 2018. DOI: https://doi.org/10.1016/j.autcon.2017.09.010

[14] VUFORIA. **Developer Library.** 2019. Disponível em: https://library.vuforia.com/articles/Training/VuMark. Acesso em: 23 jun 2019.