

Trabalho final de Sistemas em Tempo Real - Aplicações BIM em Tempo Real

Pedro Arantes Mendonça Toledo Almeida -
11621ECP008

Dezembro de 2018

1 - Introdução

BIM, ou *Building Information Modeling* é um processo que envolve a geração e manuseio de representações digitais de características físicas e funcionais de lugares. BIMs então se tornaram formatos com dados não-proprietários que podem ser trocados para apoiar decisões de construção de edifícios ou outros tipos de construção. Atualmente softwares BIM são usados por indivíduos, negócios e agências governamentais que querem planejar, construir, operar e manter diversas infraestruturas físicas como tratamento de água, dutos de gas, estações elétricas, estradas, rodovias, ferrovias, portos e túneis.^[1]

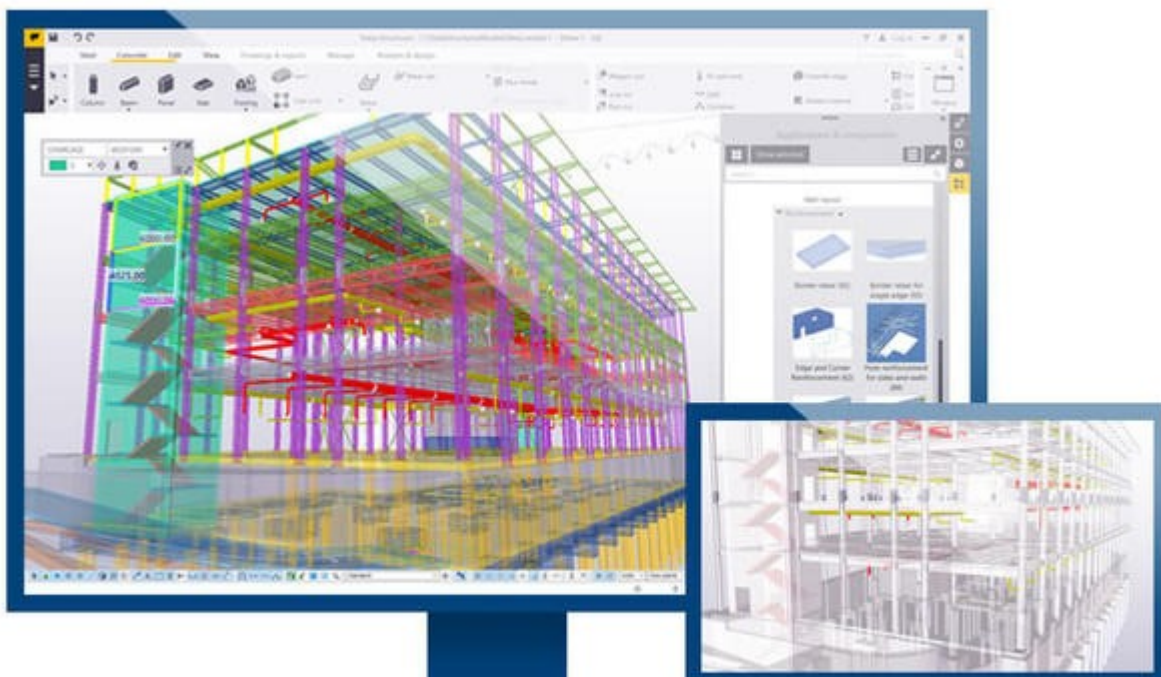




Figura 1: Exemplo de modelo BIM para projetos estruturais.

Desde sua concepção nos anos 70 até os dias de hoje, o formato BIM tem **crescido**, principalmente devido ao fato de que os governos países como Estados Unidos, Austrália, Alemanha, Finlândia, Dinamarca e Coreia do Sul estão focados em incrementar os aspectos tecnológicos da indústria de construção, principalmente nos setores de infraestrutura. Assim como o CAD possibilitou a automação da criação de informação, o BIM possibilita a automação do uso da informação, portanto o uso de BIM tanto nos setores privados quanto públicos tende a crescer.

No entanto até o momento, as aplicações do BIM se encontram muito limitadas em aplicações em tempo real, devido em parte ao fato de que os softwares de uso dos metadados associados aos modelos costumam ser de custo **extremamente elevado**, e não são softwares apropriados para renderização em tempo real. Tornando assim sua aplicação para simulações de projeto, ou projetos que continuam a ser observados e monitorados, após sua criação, até o momento impossível.

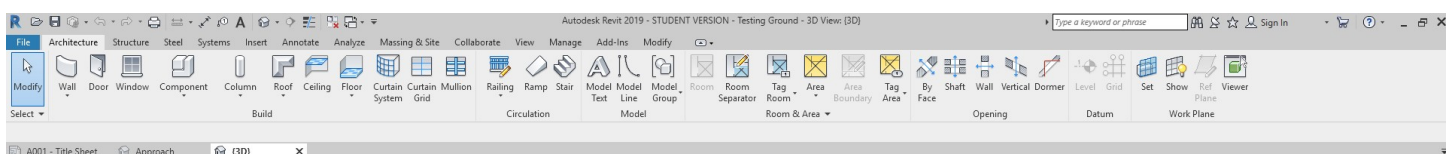
Surge então a motivação do projeto, que é de criar um workflow no qual seja possível usar de ferramentas para a geração de ambientes 3D voltados para aplicações em tempo real que contenham os mesmos metadados que um modelo BIM possui.

2 - Desenvolvimento

Primeiramente é feito um levantamento e seleção dos softwares a serem usados. Sendo eles:

- **Revit:** Software da Autodesk voltado a geração de modelos BIM.
- **Unity Engine:** *Game Engine* voltada para aplicações 3D em tempo real.
- **Dynamo:** Software open source de scripting visual voltado para BIM.

Foram escolhidos Revit e Unity pois são respectivamente os softwares mais difundidos para criação de modelos BIM, e criação de ambientes virtuais renderizados em tempo real. Para propósito de teste do workflow a ser proposto, será usado a amostra de projeto BIM que já se encontra presente no Revit.



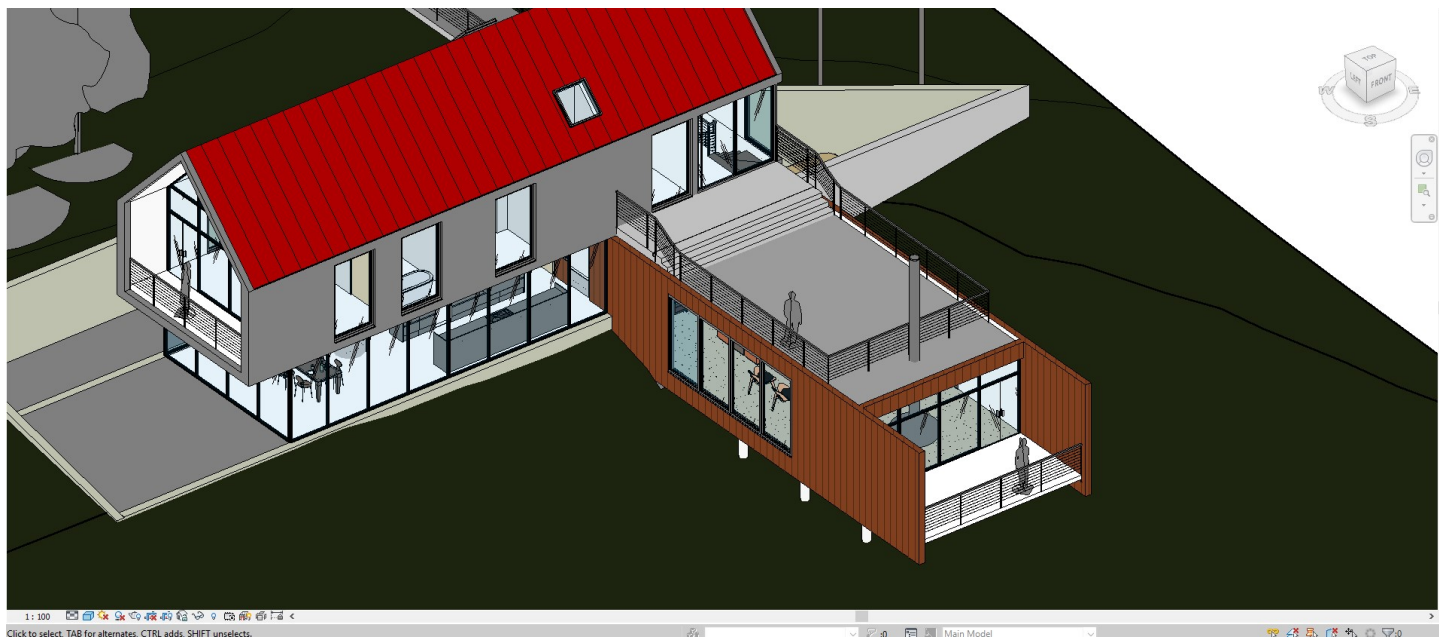


Figura 2: Modelo adotado para o desenvolvimento do projeto.

Na tentativa inicial de se passar o modelo BIM visto na Figura 2 para a Unity Engine, se depara com o maior dos problemas que ainda atrasam a adoção do BIM como padrão, o problema da **interoperabilidade**. Para que seja possível a passagem da geometria do modelo BIM, ele deve ser convertido para o formato mais comum entre aplicações 3D em tempo real: o **FBX**^[2].

O FBX é, no momento, um padrão de middleware visando a interoperabilidade entre aplicações em 3D. Sendo usado para transferir informações sobre a geometria dos modelos entre aplicações distintas. O maior problema se encontra exatamente no fato de que apenas a geometria é transferida, e não os dados BIM que os tornam um grande avanço sobre outros modelos. Sem esses metadados, os modelos BIM se tornariam iguais ou até mesmo inferiores em relação a outros formatos na quantidade de informação relevante em aplicações em tempo real devido ao fato de que todos os metadados são perdidos na transferência, e só a geometria é mantida.

Primeiramente então devemos fazer uma separação dos metadados e da geometria do modelo, e para isso primeiro iremos criar algo que associe os componentes geometricos aos seus respectivos metadados, o que irá permitir a criação de um banco de dados contendo todos os metadados de todos os componentes com um identificador. O componente identificador será denominado **Element ID**. A partir disso, é criado um script visual no Dynamo que atribui esse identificador a todos os componentes da cena, tendo o resultado visto na Tabela 1.

Script visual Dynamo	Schedule	Tabela CSV
----------------------	----------	------------

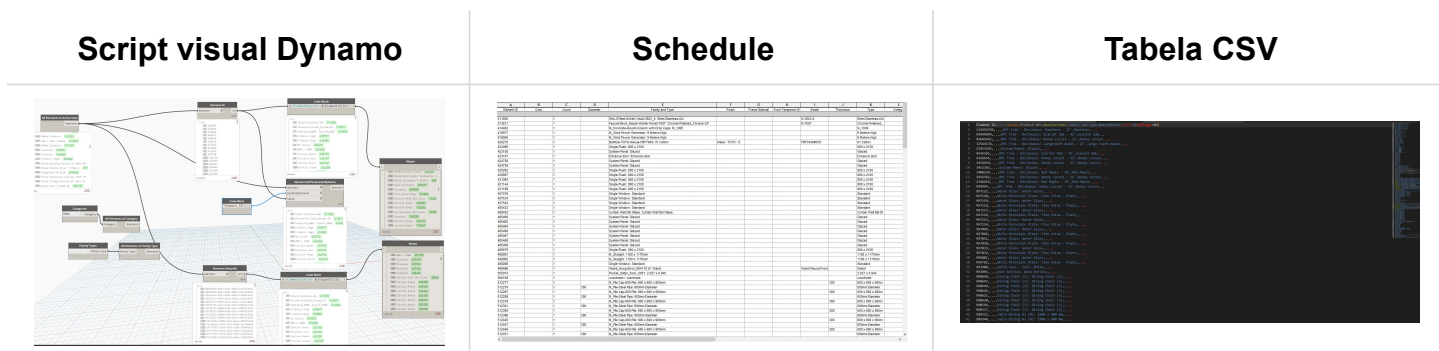


Tabela 1: Geração da tabela de banco de dados BIM.

A partir disso será criado uma **Schedule**, que nada mais é que uma tabela de banco de dados com os dados relevantes que o usuário determinar como sendo relevante para o projeto em questão. O *header* dessa tabela será o Element ID criado pelo script visto acima, e após essa Schedule ser exportada como arquivo CSV, é exportado também a geometria como se faria normalmente, no formato FBX.

Após ter em mãos tanto a geometria quanto a tabela CSV, passamos ambos os arquivos para uma cena na Unity Engine. Na Unity então é feito dois scripts na linguagem de programação **C#**, um que irá conter os metadados na forma de um componente, e outro que irá percorrer o arquivo CSV e atribuir aos respectivos objetos o seu conjunto de metadados BIM, usando do Element ID para identificar os objetos respectivos na cena. Após a execução desses scripts estaria então resolvido o maior dos obstáculos de se usar BIM em aplicações em tempo real que é o da interoperabilidade.

Após isso ter sido feito, finalmente é escrito mais alguns scripts que tornarão a cena um pouco mais interativa, permitindo que o operador do programa navegue a cena, e que o objeto que ele clicar tenha seus metadados visíveis e expostos na GUI. Tendo assim uma solução que permita a visualização, utilização e ultimamente manipulação dos metadados BIM em tempo real.

3 - Resultados

Após a execução de todos os passos propostos na etapa 2, foi possível obter uma cena conforme o previsto, onde todos os metadados são atribuídos como componentes de objetos em cena, e que são, de fato, manipuláveis e visíveis ao operador. Não só isso mas também foi possível obter uma ótima performance em tempo real devido ao uso do paradigma de programação que visam a otimizar cenas com diversos componentes chamado **Entity Component System**^[3].

Obteve-se os resultados esperados como pode ser visto na Figura 3, onde o usuário é capaz de clicar e expor os metadados BIM de cada objeto desde que o mesmo possua os metadados, de

acordo com as tabelas formadas na Tabela 1. Assim, torna-se possível um leque de aplicações para arquivos BIM extremamente versátil. O resultado mostrado na Figura 3 mostra uma das possíveis aplicações, que conta com sensoriamento de gás e fumaça em tempo real de um ambiente através de sensores MQ-2, e é possível verificar que alterações nas condições reais refletem em alterações simultâneas no ambiente virtual, mostrando sua aplicabilidade em tempo real.

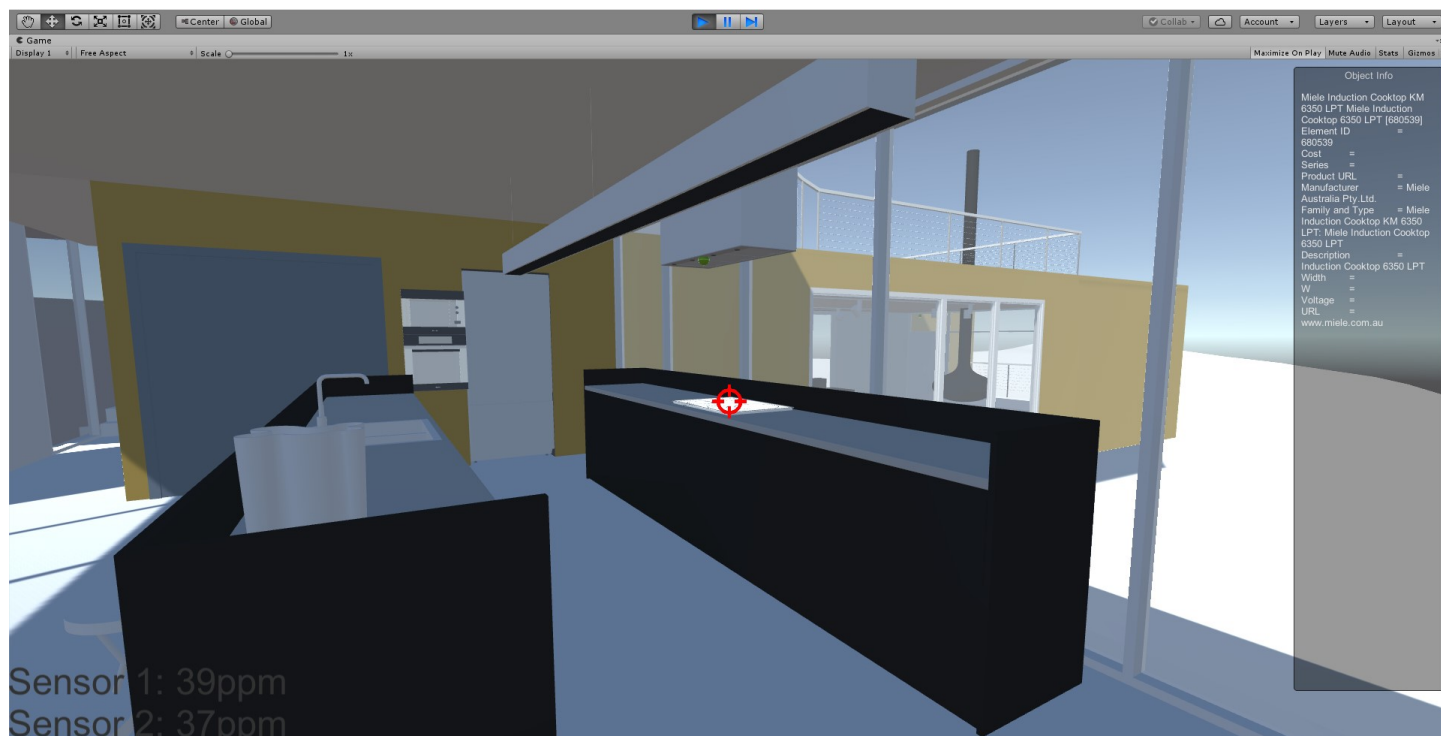
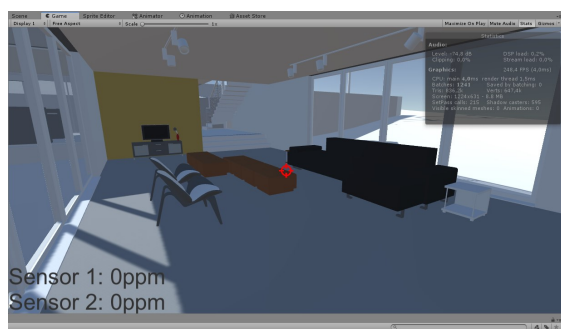


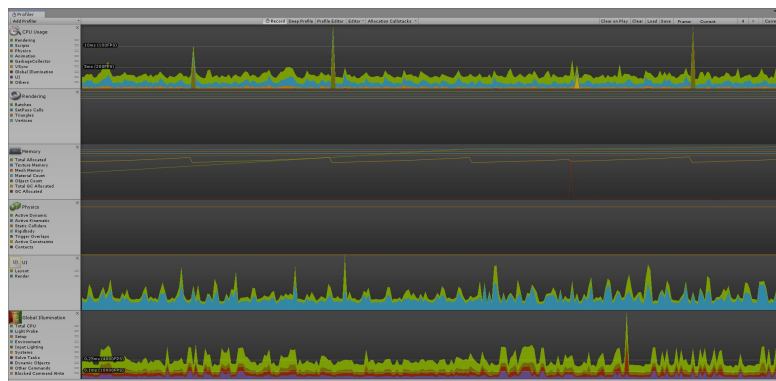
Figura 3: Resultado da cena em execução, com os metadados expostos.

Com as otimizações mencionadas acima foi possível obter ótimos resultados tratando-se de performance, mostrados na Tabela 2. Isso é extremamente importante pois, visando a maximização da usabilidade, é crucial que o software seja o mais acessível possível, portanto diminuindo o uso de recursos de hardware.

Performance em Runtime



Profiler da Performance



Performance em Runtime



Tabela 2: Análise da performance obtida em runtime

É possível notar que mesmo com um computador de usuário comum é possível executar a cena, assim é possível fazer o uso desse tipo de aplicação em tempo real de forma muito difundida, se tornando viável tanto para projetos residenciais quanto para grandes edificações.

4 - Conclusões

Foi possível, com a realização desse trabalho, a criação de um workflow e conjunto de software/hardware que torna possível contornar a grande limitação atual do BIM que é a interoperabilidade. Abrindo as portas para sua implementação em tempo real de forma ampla, sendo possível a manipulação dos seus metadados em tempo real, o que é de grande interesse nas áreas de construção, visualização da informação, simulação, entre outros. Assim torna-se possível a simulação em tempo real com a utilização de metadados BIM, que antes era um processo restrito, extremamente lento e custoso, normalmente sendo levado para outros softwares proprietários como o SolidWorks, e tinham um escopo limitado.

Pela natureza genérica de Game Engines como Unity e Unreal, torna-se possível a utilização de uma gama extremamente vasta de projetos de simulação e visualização, e com os metadados BIM, a consistência lógica de simulações estruturais seria possível de uma maneira muito mais veloz, versátil e ampla. Podendo simular de ambientes residenciais e comerciais, a modelos de máquinas e veículos.

Para a prova de conceito, foi possível notar a facilidade com a qual é possível integrar essas ferramentas com as de monitoramento em tempo real, como é o caso dos sensores MQ-2, e que assim é possível criar até mesmo redes de sensores para monitorar e simular ambientes reais permitindo uma visão computacional de situações reais muito amplificada, pois sabendo as características físicas dos projetos modelados em BIM, e o estado atual do projeto com ferramentas de monitoramento como sensores e câmeras, seria possível simular as condições de um projeto em tempo real, o que novamente abriria mais possibilidades para trabalhos futuros.

5 - Referências

[1]: Eastman, Charles; Fisher, David; Lafue, Gilles; Lividini, Joseph; Stoker, Douglas; Yessios, Christos (September 1974). An Outline of the Building Description System. Institute of Physical Planning, Carnegie-Mellon University.

[2]: FBX Product Overview. **Autodesk**, 1. Dezembro. Disponível em: <https://www.autodesk.com/products/fbx/overview>. Acesso em primeiro de Dezembro. 2018.

[3]: MARTIN, Adam. Entity Systems are the Future of MMOG Development. **T-Machine**, 1. Dezembro. Disponível em: <http://t-machine.org/index.php/2007/09/03/entity-systems-are-the-future-of-mmog-development-part-1/>. Acesso em primeiro de Dezembro. 2018.

1. 1 ↩

2. ↩

3. ↩