



ONTOLOGIA BIM: POR UMA POLÍTICA DE ARQUITETURA COGNITIVA

BIM ONTOLOGY: TOWARD A COGNITIVE ARCHITECTURAL POLICY

ONTOLOGIA BIM: POR UNA POLÍTICA DE ARQUITECTURA COGNITIVA

 <https://doi.org/10.56238/levv17n58-069>

Data de submissão: 25/02/2026

Data de publicação: 25/03/2026

José Luis Menegotto

Doutor em Arquitetura e Urbanismo pelo Programa de Pós-Graduação em Arquitetura (PROARQ)

Instituição: Universidade Federal do Rio de Janeiro (UFRJ)

E-mail: jlmenegetto@poli.ufrj.br

Lattes ID: 4205576885019501

Orcid: 0000-0003-4872-2574

RESUMO

O artigo descreve a metodologia para criação de um programa construtor de ontologias em OWL. O sistema utiliza dois elementos principais: planilhas Excel, para organizar as classes, propriedades e instâncias de um domínio BIM e um mecanismo de processamento programado em C# (Revit API 5.0) que lê as planilhas e gera arquivos OWL na sintaxe Manchester e TTL turtle. O trabalho revisita conceitos das décadas de 80 e 90, destacando a importância das ontologias para integrar dados BIM à Web Semântica, dentro do paradigma Open Linked Data. Defende-se a formulação de uma política de arquitetura cognitiva de processos, com os órgãos normativos como centros de difusão de conhecimento, sugerindo-se a reformulação dos modos de publicação das normas técnicas. Por fim, perante o avanço de agentes de IA, o artigo reforça a importância de investir em estruturação do conhecimento fático.

Palavras-chave: Dados Abertos e Vinculados. Web Semântica. Ontologia. Normas.

ABSTRACT

The article describes the methodology for creating an ontology-building program in OWL. The system uses two main elements: Excel spreadsheets, which organize classes, properties, and test instances in a BIM domain, and a processing mechanism programmed in C# (Revit API 5.0). This mechanism reads the spreadsheets and generates OWL files in Manchester syntax and TTL turtle. The work revisits concepts from the 1980s and 1990s, highlighting the importance of ontologies for integrating BIM data into the Semantic Web, within the Open Linked Data paradigm. It advocates for the formulation of a cognitive architecture policy for processes, with regulatory bodies acting as centers for knowledge dissemination, suggesting a reformulation of the ways technical standards are published. Finally, in view of the advance of AI agents, the article reinforces the importance of investing in the structuring of factual knowledge.

Keywords: Open Linked Data. Semantic Web. Ontology. Regulatory Bodies.

RESUMEN

Este artículo describe la metodología para crear un programa generador de ontologías en OWL. El sistema utiliza dos elementos principales: hojas de cálculo de Excel para organizar las clases, propiedades e instancias de un dominio BIM, y un mecanismo de procesamiento programado en C# (API de Revit 5.0) que lee las hojas de cálculo y genera archivos OWL con sintaxis Manchester y Turtle TTL. El trabajo retoma conceptos de las décadas de 1980 y 1990, destacando la importancia de las ontologías para integrar datos BIM en la Web Semántica, dentro del paradigma de Datos Abiertos Enlazados. Aboga por la formulación de una política de arquitectura cognitiva para los procesos, con los organismos reguladores como centros de difusión del conocimiento, sugiriendo una reformulación de los métodos para la publicación de estándares técnicos. Finalmente, dado el avance de los agentes de IA, el artículo refuerza la importancia de invertir en la estructuración del conocimiento fáctico.

Palabras clave: Datos Abiertos y Enlazados. Web Semántica. Ontología. Estándares.

1 INTRODUÇÃO

Desde o final de 2022, com a difusão e o uso dos novos sistemas de Inteligência Artificial generativa, como o ChatGPT ou o Midjourney, foram renovadas as especulações sobre a proximidade do momento da Singularidade Tecnológica, tal como visionada por Vernor Vinge no início da década de 1990. Ele imaginou um momento de inflexão no qual a humanidade entraria em um processo de aberta expansão da inteligência, evoluindo rumo a uma superinteligência. Segundo Vinge, a Singularidade seria um destino irreversível, que ocorreria graças aos desenvolvimentos em Inteligência Artificial ou ao desenvolvimento de redes de Inteligência Ampliada. Na década de 1990, Vinge imaginava que o momento da Singularidade se daria em uma janela temporal que aconteceria não antes de 2005 e não depois de 2030 (Vinge, 1993). A coincidência cronológica entre as visões de Vinge e o lançamento midiático dos recentes sistemas generativos reatualizou o debate.

Como contraponto, Landgrebe e Barry Smith (2023) apresentaram seus argumentos contrários à Teoria da Singularidade e à Era da inteligência pós-humana. Eles afirmam que a confiança no surgimento de uma Inteligência Artificial Geral é excessiva e que o aparecimento de um salto tecnológico como o da Singularidade seria impossível. Ambos entendem a inteligência como um produto emergente do funcionamento de um sistema complexo, observando que a largura de banda necessária para o processamento e a comunicação eficaz de uma superinteligência teria de superar os obstáculos da realidade física. Além disso, acrescentam que ainda faltam modelos matemáticos que permitam a modelagem de sistemas de Inteligência Artificial Geral. Os argumentos são significativos, pois provêm de pesquisadores reconhecidos dentro desse campo disciplinar. As pesquisas de Smith, junto com Pierre Grenon, resultaram em uma ontologia de alto nível denominada BFO (Basic Formal Ontology), presente no complexo ontológico OBO Open Bio-Ontology, amplamente adotado no âmbito das ciências biológicas. Não obstante, é indiscutível que a Inteligência Artificial vem impactando profundamente o modo de conhecer e de pensar todo o processo projetual. Neste trabalho, entende-se que o ato de projetar é um ato de conhecimento (Menegotto, 2023:1) e, portanto, um ato cognitivo humano que pode expressar processos considerados inteligentes — e que tais processos poderiam ser artificialmente assistidos. Parte-se da suposição de que os projetistas conhecem os motivos e os aspectos relevantes que levaram à formalização geométrica e física do objeto projetado, sendo capazes de transmitir esses motivos de maneira oral, textual ou gráfica, por meio de explicações que transitam por diversos níveis de tecnicidade. Destaca-se também que esses conhecimentos permanecerão implícitos no corpo do objeto construído ou projetado. Por isso, considera-se necessário utilizar modos de expressar, em linguagem formal e computacional, os conhecimentos humanos incorporados ao objeto projetado, preservando-os e, ao mesmo tempo, integrando-os aos processos tradicionais de transmissão de conhecimentos. Em outras palavras, seria necessário explicitar a teoria do objeto projetado, a fim de orientar outros agentes (humanos e algorítmicos) sobre como aproveitar

os aspectos essenciais daquilo que foi pensado e cristalizado na forma de objeto construído. Essa é uma necessidade tecnológica aberta desde o surgimento das técnicas de formalização ontológica dos sistemas atuais.

As ontologias são peças-chave para realizar esse processo, embora ainda sejam recursos pouco explorados no campo do projeto e da construção. Ontologias extensíveis para complementar modelos de dados CAD BIM vêm sendo estudadas com maior atenção nos últimos anos. Além de permitir enriquecer a semântica dos projetos (Al-Hakam et al., 2020), também possibilitam incorporar meios para mecanizar operações de raciocínio lógico nas aplicações, tornando-as mais eficientes para assistir os projetistas. Nesse campo, podem-se destacar alguns trabalhos pioneiros, como BOT (*Building Topology Ontology*), que propõe uma axiomática para BIM (Rasmussen et al., 2017); o projeto BROT (*Bridge Topology Ontology*), baseado no BOT, mas orientado a projetos de infraestrutura urbana; o projeto *Brick*, que propõe ordenamentos semânticos para sensores e dispositivos IoT em edificações; o projeto SAREF (*Smart REFerence Ontology*) (Poveda-Villalón et al., 2018); e a iniciativa do projeto ifcOWL, que busca incorporar a camada ontológica faltante ao IFC (*Industry Foundation Classes*) da organização BuildingSmart (Bonduel et al., 2018). O construtor apresentado se soma a essas iniciativas, propondo um método que ajude projetistas e organizações a formalizar ontologias. Algumas linguagens de programação utilizadas em aplicações de projeto computacional, como o DesignScript, são reconhecidas por permitir a evolução do fluxo de pensamento dos projetistas durante o processo de concepção (Aish, 2013). Mas não estão preparadas para expressar e manter um registro do conhecimento factual ordenado que orientou a concepção do objeto projetado. Essa função pode ser assumida por linguagens que permitam a formalização ontológica, ampliando assim o campo do pensamento implicitamente incorporado ao projeto no momento da concepção, pois se expande a possibilidade de expressar, de modo lógico e computacional, as determinações factuais pensadas para o objeto. Como a formalização de ontologias extensíveis é uma atividade que tem sua origem na ciência da computação, persiste certa dificuldade em envolver os projetistas no processo de modelagem de bases de conhecimento orientadas ao CAD e ao BIM que se sustentem por axiomas ontológicos. Em geral, os projetistas recebem os esquemas de informação já preparados e se adaptam a eles. O projetista conhece os objetos projetuais, conhece os comandos das aplicações CAD ou BIM, conhece alguns modos de automatizar processos, mas tem dificuldade para explicitar os conhecimentos acumulados durante as diversas etapas de concepção e modelagem, pois montar um sistema ontológico sobre o projeto não é um processo trivial (Bus et al., 2018). Além de exigir um esforço de teorização de conceitos universais e particulares, essa tarefa também depende de fatores que transcendem a escrita da ontologia e que estão relacionados com a construção coletiva de uma infraestrutura digital de conhecimentos baseada na Web Semântica que ainda é inexistente.

Escrever uma ontologia implica realizar uma reflexão contínua que conduza desde um conjunto original de conhecimentos factuais de determinações construtivas até um conjunto de novos conhecimentos factuais. Entre ambas as margens do processo estaria o projeto, que se deve servir da infraestrutura de conhecimentos posicionada nessas duas margens. Construir uma ontologia significa teorizar não apenas sobre como se projeta, mas também sobre como orientar as ferramentas de trabalho (sistemas arquitetônicos, sistemas legais, sistemas industriais, sistemas comerciais e sistemas informáticos) para que possam processar as camadas implícitas de conhecimento que vão sendo construídas durante o ato projetual e que precisariam ficar vinculadas ou incorporadas ao modelo digital. Tudo isso implica somar, às urgências práticas da tarefa projetual, os aspectos mais teóricos ou epistemológicos dessa atividade. Não muito tempo atrás, os projetos eram guardados e preservados em desenhos técnicos armazenados em mapotecas. Hoje em dia, se sustenta que é muito conveniente preservar um complexo ordenado de ontologias de cada projeto, com o conhecimento explicitamente declarado. Essa é uma atividade necessária dentro da realidade tecnológica atual.

Uma das lições aprendidas pelos autores do projeto SAREF afirma que a extração sistemática e consistente dos dados de um modelo se dificulta quando não há especialistas do domínio na equipe de projeto (Poveda-Villalón *et al.*, 2018). A essa observação pode-se acrescentar que, antes de serem sistematicamente extraídos, os dados precisam ser sistematicamente inseridos. Em outras palavras, construir ontologias orientadas à fase de concepção do projeto poderia não apenas facilitar as operações posteriores de extração de informação, mas também ajudar a materializar uma descrição ontogenética de cada projeto singular, que acompanhará o edifício durante todas as etapas posteriores, completando seu ciclo de vida. A seguir se relatam os fundamentos da aplicação que está sendo desenvolvida para facilitar a construção de ontologias por parte dos projetistas e da indústria.

2 METODOLOGIA DO CONSTRUTOR DE ONTOLOGIAS

A modelagem de ontologias é uma das vertentes de desenvolvimento no campo da Inteligência Artificial, que busca produzir aplicações com raciocínio formal incorporado, dentro da tradição da Inteligência Artificial simbólica ou top-down, inspirada no paradigma da teoria do *General Problem Solver* de Newell e Simon. A definição clássica das ontologias é devida a Gruber (1995). Ele as definiu como “uma especificação explícita de uma conceitualização”. Acrescenta que, formalmente, uma ontologia é a “declaração de uma teoria lógica”. Em termos práticos, esclarece que as ontologias compartilhadas entre agentes devem definir o vocabulário necessário para que possam ser realizadas tarefas de inteligência geral a partir de interações em interfaces do tipo *tell* and *ask* (declarar e questionar), isto é, interfaces de afirmações e consultas sobre os objetos. Gruber utiliza esses termos citando o trabalho de Levesque, que na década de 1980 abordou o problema da representação lógica do conhecimento a partir de uma perspectiva funcionalista. Levesque (1984) sugere que as operações

primitivas para formar aplicações complexas baseadas em bases de conhecimento (*Knowledge Base*) passam por duas etapas: a primeira seria a fase de aquisição do conhecimento e a segunda, o momento em que o conhecimento pode ser aumentado. Embora Levesque se limite a duas primitivas operacionais (acessar e aumentar), ele esclarece que os agentes poderiam realizar outras operações de conhecimento, como “esquecer ou assumir”. Ele esquematiza os estágios da seguinte forma:

Declara conhecimento: Declara: Conhecimento $\times$ Asserção $\rightarrow$ Conhecimento

Acessa conhecimento: Pergunta: Conhecimento $\times$ Filtro $\rightarrow$ Resposta

Aumenta conhecimento: Declara: Conhecimento $\times$ Asserção $\rightarrow$ +Conhecimento

O termo “agente” é utilizado para significar tanto agentes humanos quanto agentes algorítmicos. Outro esclarecimento importante, baseado em Gruber, é que o uso dos conhecimentos formalizados ontologicamente não estaria fundamentado em sua completude, mas em sua consistência. Em outras palavras, o conhecimento adquirido pelos agentes não precisa ser simétrico nem completo, pois cada agente pode saber aspectos diferentes sobre o mesmo objeto (Gruber, 1995). Resumindo, as ontologias são um meio computacional para expressar, em linguagem formal, o conhecimento teórico e factual que se possui sobre os objetos. Para isso, os envolvidos na criação de ontologias devem ser capazes de definir e avaliar as entidades propostas, por meio de asserções explícitas sobre classes de objetos, suas propriedades, as propriedades dos dados utilizados pelos objetos e, finalmente, as asserções de “fatos” concretos (*facts*) evidenciados pelos indivíduos conceptualizados (instâncias). A partir do conjunto de asserções e fatos explícitos, os sistemas de cálculo lógico (*reasoners*) que processam ontologias podem inferir conhecimentos não declarados (implícitos), mecanizando operações de cálculo proposicional. Continuaremos explicando um sistema que foi implementado em um programa para facilitar o desenvolvimento de módulos ontológicos em formato OWL, pensados para serem integrados em sistemas de dados abertos e conectados, de acordo com o conceito de Web Semântica de Berners-Lee (2009), para inter-relacionar sistemas e silos de informação da construção. Os módulos procuram explicitar descrições conceituais dos elementos de projetos arquitetônicos e podem ser aproveitados para ordenar e direcionar dados em ecossistemas CAD BIM durante as etapas de concepção, construção ou gestão pós-construção. A definição axiomática de uma ontologia exige transitar por diversos níveis de abstração dentro de cada domínio estudado. No domínio estrutural, por exemplo, podem ser utilizados conceitos estruturais já conhecidos e catalogados, como os propostos por Engel (2007), que permitem compreender a estrutura de um edifício a partir da perspectiva da transmissão de cargas. Para categorizar objetos devem ser considerados os diversos esquemas de dados concorrentes, estabelecidos de maneira diferente em aplicações proprietárias como Revit ou em esquemas de dados neutros como o IFC. Portanto, a inteligência que poderá emergir de um sistema complexo formado por ontologias dependerá de dois aspectos: a separação dos domínios de conhecimento e os vínculos que possam ser estabelecidos entre eles. Na aplicação apresentada, os

arquivos OWL são escritos em notações Manchester e Turtle e criados modularmente a partir de planilhas Excel gerenciadas por macros programadas em API escrita em linguagem C# para Revit. As ontologias podem ser abertas no editor Protégé para visualizar, gerenciar, depurar e validar sua estruturação. Cada módulo ontológico deve corresponder a um domínio específico e possuir um prefixo que funcione como *NameSpace* ou identificador URI (*Uniform Resource Identifier*), por exemplo, fofu: significando forma e função, ou bim:. Essa chave permite integrar as ontologias dentro de fluxos de conhecimento da Web Semântica, possibilitando a incorporação vinculada de recursos informacionais RDF no paradigma de *Open Linked Data* proposto por Berners-Lee (2005, 2009).

3 RECOMENDAÇÕES FUNDAMENTAIS

Para implementar o mecanismo, foram consideradas as cinco recomendações formais de Gruber (1995) e o realismo de Smith (2008). Gruber recomenda que uma ontologia seja clara, coerente, extensível, com mínima codificação e minimamente comprometida. Smith, por sua vez, recomenda que ela seja formulada dentro de uma perspectiva realista.

A primeira recomendação de Gruber, a *clareza*, exige que as definições conceituais das entidades sejam, na medida do possível, necessárias e suficientes. O *realismo* de Smith implica que as asserções declaradas sobre cada entidade sejam feitas com base em fundamentos objetivos, isto é, distinguindo a representação da entidade real (objeto) das representações que carreguem algum viés derivado de uma ação cognitiva humana, ou seja, de interpretações subjetivas. Esse é um desafio para as ontologias de concepção de projeto, pois a própria concepção projetual é, em si, uma sequência de ações que implicam atos cognitivos humanos. O mecanismo programado buscou facilitar o esforço conceitual, permitindo a rápida reconfiguração das classes, propriedades e relações pensadas para construir os conceitos axiomáticos, além de expressar cada conceito em linguagem natural. Na primeira versão do mecanismo, a expressão dos termos foi limitada ao idioma português, utilizando o axioma “*Annotations*” em cada linha da planilha e para cada nível da árvore de classes. Na versão atual, foram incorporadas as explicações dos conceitos em inglês e espanhol.

O segundo aspecto, a *coerência*, refere-se ao vínculo entre a descrição formal e as descrições informais que podem ser feitas em linguagem natural. Estas não podem ser contraditórias. As cores escolhidas para as células da planilha Excel buscam facilitar a visualização global do campo ontológico, ajudando a identificar possíveis sobreposições de conceitos contraditórios. As cores utilizadas são as mesmas empregadas pelo programa Protégé, que foi criado na Universidade de Stanford e é amplamente utilizado para a criação de ontologias:

- **Amarelo** para as classes
- **Azul** para as propriedades de objetos
- **Verde** para as propriedades de dados

- **Magenta** para os indivíduos instanciados
- **Ocre** para as anotações

A terceira recomendação, a *extensibilidade*, significa que a definição de um novo conceito possa ser herdeira “monotônica” dos conceitos anteriores. Gruber usa essa expressão para indicar que os membros menores devem estar englobados em conjuntos de igual gênero ou natureza, permitindo que os conceitos possam crescer sem afetar a estruturação da árvore ontológica e do vocabulário previamente definido. O ordenamento das células B, C e D e E da tabela 2 segue essa recomendação. Essa condição reduziria os esforços de adaptação ou redefinição das terminologias propostas. A quarta recomendação pede utilizar um *mínimo de viés de codificação*. A escolha da sintaxe Manchester (Horridge *et al.*, 2004), cuja escrita é quase em linguagem natural, fundamenta-se nisso. Manchester utiliza uma quantidade mínima de caracteres especiais (símbolos) em comparação com outras sintaxes usadas para o mesmo fim como XML ou JSON. Exemplo de classe com anotações e asserções de equivalência escrita em Manchester:

Class: bim:Ambiente

SubClassOf: bim:Programa.de.Necessidades

Annotations: rdfs:label "bim:Ambiente"@pt

Annotations: rdfs:comment "Programa de Necessidades"@pt

EquivalentTo: bim:OST_Rooms or bim:IfcSpace

Na versão atual do construtor foi incorporada a sintaxe Turtle, que embora seja levemente mais codificada ainda mantém uma leitura natural. Finalmente, para alcançar um *compromisso ontológico mínimo*, busca-se conceituar os objetos da maneira mais simples possível. Essa tarefa está no núcleo da conceitualização dos objetos projetuais que os ontologistas precisam realizar. Ela está relacionada à clareza e ao realismo, constituindo o grande desafio epistemológico do projeto. É importante destacar que as recomendações de Gruber e Smith têm origem nos campos disciplinares das ciências cognitivas e da computação que, com o objetivo de incorporar mecanismos de raciocínio em sistemas computacionais, concebem linguagens que permitem realizar operações de lógica matemática ou de primeira ordem, cuja matéria-prima é composta por um número indeterminado de premissas (asserções e fatos) e um conjunto limitado de operações lógicas. Como resultado, o aplicativo gera arquivos OWL e TTL, legíveis por algoritmos e ainda de fácil leitura para agentes humanos. Esses arquivos podem ser importados no Protégé para verificar sua coerência formal.

4 AS PLANILHAS

O algoritmo construtor processa seis planilhas de um arquivo Excel, que contêm a descrição dos aspectos do domínio formalizado. A versão mais recente do construtor é a Versão 5.0. O

código-fonte, as planilhas e os arquivos OWL e TTL resultantes podem ser consultados e baixados para testes em:

<https://github.com/JLMenegotto/OntologiaBIM/tree/main/Vers%C3%A3o5>

4.1 ANOTAÇÕES GERAIS DO PROJETO

Em duas colunas (Tabela 1) são declarados os aspectos gerais de um projeto: endereço, tipo de construção e posicionamento geográfico são algumas das anotações possíveis. Na primeira coluna declara-se o nome do parâmetro e, na seguinte, o seu valor. Como cada arquivo Excel serve a um domínio específico, a primeira linha deve definir o NameSpace da ontologia. O NameSpace é fundamental, pois permite a incorporação modular da ontologia em ambientes que utilizam recursos RDF dentro do paradigma de Dados Abertos e Conectados proposto por Berners-Lee (2009), para formar a rede de recursos informacionais vinculados na Web Semântica. As anotações dos dados podem ser ampliadas à medida que a equipe adquire novos conhecimentos relativos ao projeto. A terceira coluna é reservada para declarar a linguagem da declaração.

Tabela 1. Anotações de propriedades gerais do projeto.

	A	B	C
1	Chave	Valor	Idioma
2	OntologiaPrefixo	bim:	pt
3	OntologiaTema	Parte_2_OM	pt
4	RaizClass	BIM	pt
5	RaizProp	BIMProp	pt
6	RaizData	BIMData	pt
7	URI	https://github.com/JLMenegotto/RepoOnto#	pt
8	Autor	Prof. José Luis Menegotto - DEG/POLI/UFRJ	pt
9	Organização	Escola Politécnica da UFRJ	pt
10	NormaNúmero	NBR15965-2:2012	pt
11	Edição	v-2012	pt
12	ISBN	978-85-07-03497-1	pt
13	NormaEscopo	Parte 2 - OM: Características dos objetos Materiais	pt
14	NormaReferência1	-	pt
15	NormaReferência2	-	pt
16	NormaReferência3	-	pt
17	Observações	Sistema de classificação ABNT da construção para BIM	pt
18	DataHora	04/02/2026 14:32	pt
19	Advertência	A Norma foi elaborada pela Comissão de Estudo Especial de Modelagem da Informação da Construção (BIM) ABNT/CEE-134	pt
20	Aviso	El Estándar fue preparado por el Comité Especial de Estudio sobre Modelado de Información de Edificación (BIM) ABNT/CEE-134	es
21	Alert	The Standard was prepared by the Special Committee of Studies on Building Information Modeling (BIM) ABNT/CEE-134	en
22	Prédio	-	pt
23	Cidade	-	pt
24	Explicação	Formalizar a Norma brasileira NBR15965 no domínio do Sistema de classificação ABNT da construção.	pt
25	Explicación	Formalizar la Norma brasileira NBR15965 en el dominio del Sistema de clasificación de la construcción.	es
26	Explanation	Formalize the Brazilian Standard NBR15965 within the domain of the Construction Classification System	en

Fonte: o autor

4.2 CLASSES, ASSERÇÕES, EQUIVALÊNCIAS E ANOTAÇÕES

Uma planilha declara a estrutura hierárquica das classes (Tabela 2). As cinco primeiras colunas, iniciando pela coluna B, definem uma árvore hierárquica. A raiz da árvore é a classe da coluna B, subindo por quatro níveis: SuperClasse2 → SuperClasse3 → SuperClasse4 → até a Classe na coluna F. Esta coluna não pode ter classes repetidas. Todas as classes são escritas em maiúscula e quando precisar de mais de uma palavra para nomeá-la se utiliza um ponto como separador. As ontologias são integradas e interconectadas modularmente dentro de um complexo ontológico no qual cada uma declara fatos conhecidos sobre os objetos específicos de um domínio em um determinado nível de

generalidade. A BFO (Basic Formal Ontology), proposta pelo grupo de pesquisa de Smith (2008), é uma ontologia compacta e de alto nível que pode servir como raiz para entidades em outras ontologias. Observou-se que, entre as ontologias analisadas em domínios relacionados ao BIM, nenhuma está vinculada à BFO. Por razões de interoperabilidade, acredita-se que se poderia integrar a BFO como uma camada fundamental para ontologias em domínios relacionados ao projeto arquitetônico. Se os ontologistas optarem por derivar sua ontologia usando a BFO como estrutura de generalização, a raiz do conceito (Coluna B) deve ser reservada para indicar a natureza da classe à qual a entidade pertence nesse domínio. Na BFO, objetos espaciais são chamados de *Continuant*, enquanto conceitos relacionados a eventos temporais são chamados de *Occurrents*.

Tabela 2. Mapa semântico de equivalências entre Conceitos (Classes)

A	B	C	D	E	F	G	H	I	J	K
0	00Raiz	00Super Class 2	00Super Class 3	00Super Class 4	00 Super Class 5	EquivalentTo: Raiz Condições necessárias	EquivalentTo: Classe2 Condições necessárias	EquivalentTo: Classe3 Condições necessárias	EquivalentTo: Classe4 Condições necessárias	EquivalentTo: Classe5 Condições necessárias
2	Projeto	Fronteira	Geográfica	Natural	Continente	é.nome some Projeto	é.nome some Fronteira	null	null	é.nome some Continente
3	Projeto	Fronteira	Geográfica	Natural	SubContinente	null	null	null	null	null
4	Projeto	Fronteira	Geográfica	Natural	Bioma	null	null	null	null	null
5	Projeto	Fronteira	Geográfica	Política	País	null	null	null	null	é.dentro.de some Continente
6	Projeto	Fronteira	Geográfica	Política	Nação	null	null	null	null	null
7	Projeto	Fronteira	Geográfica	Política	Estado	null	null	null	null	é.dentro.de some País
8	Projeto	Fronteira	Geográfica	Política	UF	null	null	null	null	null
9	Projeto	Fronteira	Geográfica	Política	Provincia	null	null	null	null	é.dentro.de some Nação
10	Projeto	Fronteira	Geográfica	Política	Cidade	null	null	null	null	é.dentro.de some Estado
11	Projeto	Fronteira	Geográfica	Política	Metrópoli	null	null	null	null	é.dentro.de some Estado
12	Projeto	Fronteira	Geográfica	Política	Município	null	null	null	null	é.dentro.de some UF
13	Projeto	Limite	Térmico	Bioclimático	Região.Climática	null	null	null	null	é.dentro.de some Bioclimático
14	Projeto	Limite	Térmico	Bioclimático	Zona.Climática	null	null	null	null	é.dentro.de some Bioclimático
15	Projeto	Limite	Urbano	Ilha.de.Calor	Zona.Climática.Local	null	null	null	null	é.dentro.de some Ilha.de.Calor

Fonte: o autor

A folha de classes define a estrutura hierárquica dos conceitos e também as suas condições existenciais. As classes devem ser definidas começando com letra maiúscula e com termos separados por pontos. A coluna F não pode conter classes repetidas; ela possui uma regra para verificar e destacar duplicidades, que devem ser corrigidas. As colunas G a K são utilizadas para inserir condições existenciais que definem a universalidade ou a particularidade de um conceito. Os operadores disponíveis para condições existenciais são:

Some, only, or, and, max, min, exactly e not

Exemplo 1: A classe *Red*, subclasse de *Canal*, possui uma condição existencial definida por um axioma que utiliza a propriedade de objeto *é.red*, a qual limita sua configuração válida entre os valores mínimos e máximos 0 e 255.

é.red min 0, é.red max 255

Exemplo 2: O axioma aplicado à classe *Barrio* (coluna F) possui uma condição na coluna K:

é.dentro.de some Ciudad

Essa condição existencial pode ser interpretada da seguinte forma: um bairro está dentro de alguma cidade. As condições podem ser escritas concatenando conjunções (*and*) ou disjunções (*or*).

Essas afirmações agrupadas entre parênteses devem ser escritas deixando um espaço em branco após cada elemento. Por exemplo:

(*prop.ob1 some Classe*) and (*prop.ob2 some Classe*)

Durante a programação do construtor, buscou-se facilitar a escrita da ontologia eliminando a necessidade de escrever o *Namespace* antes dos nomes de classes e propriedades. Mas nos arquivos OWL e TTL gerados, cada classe ou propriedade é precedida pelo *Namespace* da ontologia, escrito automaticamente pelo construtor. (exemplo: *bim:Continente*). Isso permite que o usuário se concentre na conceituação das classes e propriedades e menos nas formas sintáticas.

4.3 MAPA SEMÂNTICO. EQUIVALÊNCIAS DE CONCEITOS

Uma planilha especial (Tabela 3) é preparada para conter um mapa de correspondências semânticas. Ela permite declarar as equivalências entre diversos modelos de dados concorrentes ou conceitos afins. Por exemplo, entre as categorias IFC e as categorias OST do Revit. Ou como no exemplo da tabela 3 com alguns significados afins de elementos de planejamento e infraestrutura urbana. Todas as classes declaradas nesta planilha devem ter sido previamente declaradas na planilha de Classes. Na coluna B são listadas todas as categorias de um determinado esquema de dados; nas colunas seguintes (C em diante) são usadas para expandir equivalências com classes equivalentes do mesmo gênero. Sugere-se que a primeira equivalência seja a mais próxima ou direta. Como os modelos de dados costumam apresentar ambiguidades ou lacunas semânticas, os ontologistas podem complementar cada linha com equivalências menos fortes, se necessário.

Tabela 3. Mapa semântico de equivalências entre Conceitos (Classes).

A	B	C	D
1	Classe	Equivalente a	Equivalente a
2	Lote	Parcela	null
3	País	Nação	null
4	Estado	UF	Província
5	Cidade	Metrópoli	Município
6	Região.Climática	Zona.Climática	null
7	Doca.Flutuante	Dolfin	null
8	Poste.Luz.Viário	OAE.Iluminação	null
9	Estação.Central	Estação.Terminal	null
10	Central.Rodoviário	Terminal.Rodoviário	null
11	Ponto.de.Ônibus	Parada.de.Ônibus	null
12	Drone	Vant	UAS
13	Trem.de.Alt.Velocidade	Trem.Bala	null

Fonte: o autor

4.4 CLASSES DISJUNTAS

Em outra planilha (Tabela 4) podem ser definidas as disjunções de classes. Por exemplo, a Classe PXYZ disjunta da Classe PGEO (posição cartesiana XYZ vs. posição geográfica Lat-Long). Categorias IFC e OST são declaradas disjuntas, já que os modelos de dados do Revit e do IFC não são

congruentes. Na ontologia de fabricantes, por exemplo, foram diferenciadas como disjuntas as classes *Fornecedores* (empresa fabricantes ou vendedoras) e *Fornecidos* (os produtos).

Tabela 4. Classes disjuntas.

A	B	C	D
1	Disjunta 1	Disjunta 2	Disjunta 3
2	Fornecedores	Fornecidos	null

Fonte: o autor

5 ARQUIVO DE PROPRIEDADES

O arquivo *Ontologia_Propriedades.xlsx* (Tabelas 5a e 5b) organiza e centraliza todas as propriedades criadas para o complexo do domínio ontológico estruturado. Isso não significa que todas as ontologias precisem incluir todas as propriedades. Cada domínio utilizará, a seu critério, apenas aquelas que forem necessárias à medida que novos conhecimentos surgirem. Novas propriedades podem ser adicionadas conforme a necessidade de declarar conhecimento factual sobre cada tema. O arquivo Excel contém a estrutura de propriedades de objetos e de dados de um complexo ontológico constituído para todos os domínios. Ele é atualizado regularmente por meio da incorporação de novas propriedades, da modificação das características de propriedades existentes ou do reagrupamento de uma propriedade dentro de outra *SubProp1* e *SubData1* (colunas C e F). A planilha deve ser preenchida a partir da coluna *G* (*SubData2*) com o texto em minúsculas e separando cada palavra da propriedade com um ponto. Quando a propriedade tiver mais de uma palavra, elas não devem ser separadas por espaços em branco. A coluna *F* (*SubData1*) define a natureza da propriedade por meio de um verbo que deve começar com o prefixo "d.".

Tabela 5a. Mapa semântico das propriedades de objetos e dados.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
Nº	Raiz	SubProp1	SubProp2	Raiz	SubData1	SubData2	Proprie	Func (9)	Func	IFunc	Transit	Sym	Asym	Reflex	Ireflex	Invol	Equiv	Dom (19)	Rng
389	BIMProp	p.pintar	d.alla	BIMData	d.pintar	alla	xsd:string	null	null	null	null	null	null	null	null	null	null	BIM	Projeto
390	BIMProp	p.pintar	d.cyan	BIMData	d.pintar	cyan	xsd:string	null	null	null	null	null	null	null	null	null	null	BIM	Projeto
391	BIMProp	p.pintar	d.magenta	BIMData	d.pintar	magenta	xsd:string	null	null	null	null	null	null	null	null	null	null	BIM	Projeto
392	BIMProp	p.pintar	d.yellow	BIMData	d.pintar	yellow	xsd:string	null	null	null	null	null	null	null	null	null	null	BIM	Projeto
393	BIMProp	p.pintar	d.black	BIMData	d.pintar	black	xsd:string	null	null	null	null	null	null	null	null	null	null	BIM	Projeto
394	BIMProp	p.pintar	d.red	BIMData	d.pintar	red	xsd:string	null	null	null	null	null	null	null	null	null	null	BIM	Projeto
395	BIMProp	p.pintar	d.reda	BIMData	d.pintar	reda	xsd:string	null	null	null	null	null	null	null	null	null	null	BIM	Projeto
396	BIMProp	p.pintar	d.cmy	BIMData	d.pintar	cmy	xsd:string	null	null	null	null	null	null	null	null	null	null	BIM	Projeto
397	BIMProp	p.pintar	d.cmyb	BIMData	d.pintar	cmyb	xsd:string	null	null	null	null	null	null	null	null	null	null	BIM	Projeto
398	BIMProp	p.pintar	d.cmyhexa	BIMData	d.pintar	cmyhexa	xsd:string	null	null	null	null	null	null	null	null	null	null	BIM	Projeto
399	BIMProp	p.pintar	d.depois.de	BIMData	d.pintar	depois.de	xsd:string	null	null	null	Transitive	null	null	null	null	null	null	BIM	Projeto
400	BIMProp	p.planejar	d.uniao.de	BIMData	d.planejar	uniao.de	xsd:string	null	null	null	Transitive	null	null	null	null	null	null	BIM	Projeto
401	BIMProp	p.planejar	d.concomitante.a	BIMData	d.planejar	concomitante.a	xsd:string	null	null	null	Transitive	Symmetric	null	null	null	null	null	BIM	Projeto
402	BIMProp	p.planejar	d.simultaneo.a	BIMData	d.planejar	simultaneo.a	xsd:string	null	null	null	Transitive	Symmetric	null	null	null	null	null	BIM	Projeto
403	BIMProp	p.planejar	d.evento.inicial	BIMData	d.planejar	evento.inicial	xsd:string	null	null	null	null	null	null	null	null	null	null	BIM	Projeto
404	BIMProp	p.planejar	d.evento	BIMData	d.planejar	evento	xsd:string	null	null	null	null	null	null	null	null	null	null	BIM	Projeto
405	BIMProp	p.planejar	d.evento.final	BIMData	d.planejar	evento.final	xsd:string	null	null	null	null	null	null	null	null	null	null	BIM	Projeto
406	BIMProp	p.planejar	d.vida.util	BIMData	d.planejar	vida.util	xsd:string	null	null	null	null	null	null	null	null	null	null	BIM	Projeto
407	BIMProp	p.prever	d.precipitacao.dia	BIMData	d.prever	precipitacao.dia	xsd:double	null	null	null	null	null	null	null	null	null	null	BIM	Projeto
408	BIMProp	p.prever	d.precipitacao.mensal	BIMData	d.prever	precipitacao.mensal	xsd:double	null	null	null	null	null	null	null	null	null	null	BIM	Projeto
409	BIMProp	p.prever	d.velocidade.do.vento	BIMData	d.prever	velocidade.do.vento	xsd:double	null	null	null	null	null	null	null	null	null	null	BIM	Projeto
410	BIMProp	p.prever	d.direcao.do.vento	BIMData	d.prever	direcao.do.vento	xsd:string	null	null	null	null	null	null	null	null	null	null	BIM	Projeto
411	BIMProp	p.prever	d.radiacao.solar	BIMData	d.prever	radiacao.solar	xsd:double	null	null	null	null	null	null	null	null	null	null	BIM	Projeto
412	BIMProp	p.prever	d.pressao.atmosferica	BIMData	d.prever	pressao.atmosferica	xsd:double	null	null	null	null	null	null	null	null	null	null	BIM	Projeto
413	BIMProp	p.prever	d.temperatura	BIMData	d.prever	temperatura	xsd:double	null	null	null	null	null	null	null	null	null	null	BIM	Projeto
414	BIMProp	p.prever	d.umidade.relativa	BIMData	d.prever	umidade.relativa	xsd:double	null	null	null	null	null	null	null	null	null	null	BIM	Projeto
415	BIMProp	p.prever	d.qualidade.do.ar	BIMData	d.prever	qualidade.do.ar	xsd:double	null	null	null	null	null	null	null	null	null	null	BIM	Projeto
416	BIMProp	p.prever	d.cobertura.meteorologica	BIMData	d.prever	cobertura.meteorologica	xsd:string	null	null	null	null	null	null	null	null	null	null	BIM	Projeto
417	BIMProp	p.projetar	d.autor	BIMData	d.projetar	autor	xsd:string	null	null	null	null	null	null	null	null	null	null	BIM	Projeto

Fonte: o autor

todas as características. Não deve haver células vazias: sempre deve ser inserido um valor ou o valor *null* caso não se deseje definir a característica. A seguir, o significado de cada característica:

Functional: indica que a propriedade deve ter, no máximo, um valor único para cada indivíduo. Exemplo: data de nascimento. Uma pessoa tem apenas uma data de nascimento, embora essa mesma data possa ser válida como data de nascimento de outras pessoas.

Inverse Functional: indica que o valor da propriedade é unívoco entre sujeito e predicado. Exemplo: um número de CPF é único e identifica no máximo a uma pessoa. Ou seja, uma propriedade *Inverse Functional* é funcional nos dois sentidos.

Transitive: indica transitividade. Exemplo: a propriedade *é.dentro.de* é transitiva, pois, se A *é.dentro.de* B e B *é.dentro.de* C, então pode-se inferir que A *é.dentro.de* C.

Symmetric: *é.irmão.de* é uma típica propriedade simétrica: se A *é.irmão.de* B, também é válido inferir que B *é.irmão.de* A.

Asymmetric: *é.avô.de* é antissimétrica: se A *é.avô.de* B, não pode ser válido que B *é.avô.de* A. Essa característica ajuda a evitar contradições entre indivíduos distintos.

Reflexive: uma propriedade reflexiva indica que existe uma relação que também se aplica ao próprio indivíduo (exemplo: *é.igual.a*). Um elemento pode ser igual a outro, assim como é igual a si mesmo.

Irreflexive: uma propriedade irreflexiva indica que é sempre falso que um objeto se relacione consigo mesmo. Exemplo: *é.avô.de*, pois ninguém pode ser avô de si mesmo.

InverseOf: caracteriza a uma propriedade como inversa de outra. Indica que, se A se relaciona com B por meio dessa propriedade, então é válido afirmar que B se relaciona com A por meio de sua propriedade inversa. Exemplo: se A *é.acima.de* B, então B *é.embaixo.de* A).

5.2 ANOTAÇÕES DAS PROPRIEDADES

As colunas U e V são preenchidas automaticamente. A coluna W deve ser preenchida manualmente com comentários escritos em português que expliquem, de forma objetiva e concisa, o significado da propriedade. Esses comentários podem incluir referências a normas que utilizem a propriedade ou explicações sobre como ela deve receber seu valor. Essas explicações serão traduzidas para o espanhol na coluna Z e para o inglês na coluna AG por meio de uma fórmula aplicada na célula do Excel. Por exemplo:

=TRADUZIR(P20; "pt"; "es")

- Coluna X: identifica cada grupo de propriedades com um valor formado pelos quatro primeiros caracteres do grupo (coluna F), numerado sequencialmente a partir de 100. Serve para manter uma indexação de cada grupo.
- Coluna Y: reservada para desenvolvimentos futuros.

- Coluna AA: categorias de objetos do Revit associadas à propriedade.
- Coluna AB: valor da propriedade correspondente à coluna AA.
- Coluna AC: classes IFC associadas a essa propriedade.
- Coluna AD: valor da propriedade correspondente à coluna AC.
- Coluna AE: propriedade do sistema de classificação da construção da norma ABNT 15.965 associada à propriedade.
- Coluna AF: valor da propriedade correspondente à coluna AE.
- Coluna AG: explicação da propriedade em inglês.

6 INDIVÍDUOS E FATOS CONCRETOS

Se as Classes e as Propriedades de Objetos e de Dados definem o esquema abstrato dos conceitos, os fatos (Facts) descrevem realidades concretas e conhecidas dos indivíduos, utilizando as definições de classes e propriedades. A coluna B da planilha de Fatos (Tabela 7) é utilizada para declarar um indivíduo (pode ter qualquer nome, sem espaços). A coluna C define o tipo do indivíduo, que deve estar associado a alguma classe existente. As colunas seguintes (azuis ou verdes) são usadas para descrever as propriedades de cada indivíduo:

- *As azuis correspondem a propriedades de objetos.*
- *As verdes correspondem a propriedades de dados.*

Cada propriedade ocupa duas colunas: A primeira contém o nome da propriedade definido no esquema. A segunda contém o valor dessa propriedade. As propriedades de objetos (colunas azuis) relacionam as Classes com os Indivíduos ou os Indivíduos entre si, por isso são seguidas de uma coluna magenta que representa a um outro indivíduo já declarado. As propriedades de dados (colunas verdes) qualificam ou descrevem o valor de um dado para os indivíduos, são seguidas de um valor concreto (texto, número, lista, etc.).

Tabela 7. Indivíduos com definição de classe e propriedades.

Nº	Indivíduo	Indivíduo	Fato	Val	Fato	Val	Fato	Val	Fato	Val
17	Paleta.Pantone	Catálogo	null	null	null	null	descrição	"Cores padronizadas no sistema Pantone Matching System"	nome	"Paleta Cromática Pantone"
18	Pantone.116.C	Cor.Pantone	é.pertencente.a	Paleta.Pantone	é.usado.por	Paleta.DNIT	descrição	"Cor Pantone utilizado no DNIT"	cyan	0
19	Pantone.370.C	Cor.Pantone	é.pertencente.a	Paleta.Pantone	é.usado.por	Paleta.DNIT	descrição	"Cor Pantone utilizado no DNIT"	cyan	63
20	Pantone.3425.C	Cor.Pantone	é.pertencente.a	Paleta.Pantone	é.usado.por	Paleta.DNIT	descrição	"Cor Pantone utilizado no DNIT"	cyan	100
21	Sistema.Munsell	Munsell	null	null	null	null	descrição	"Sistema de cores Munsell"	null	null
22	Maltz.RP	Hue	é.pertencente.a	Sistema.Munsell	null	null	descrição	"Maltz (HUE) red-purple"	null	null
23	Maltz.P	Hue	é.pertencente.a	Sistema.Munsell	null	null	descrição	"Maltz (HUE) purple"	null	null
24	Maltz.PB	Hue	é.pertencente.a	Sistema.Munsell	null	null	descrição	"Maltz (HUE) purple-blue"	null	null
25	Maltz.B	Hue	é.pertencente.a	Sistema.Munsell	null	null	descrição	"Maltz (HUE) blue"	null	null
26	Maltz.BG	Hue	é.pertencente.a	Sistema.Munsell	null	null	descrição	"Maltz (HUE) blue-green"	null	null
27	Maltz.G	Hue	é.pertencente.a	Sistema.Munsell	null	null	descrição	"Maltz (HUE) green"	null	null
28	Maltz.GY	Hue	é.pertencente.a	Sistema.Munsell	null	null	descrição	"Maltz (HUE) green-yellow"	null	null
29	Maltz.Y	Hue	é.pertencente.a	Sistema.Munsell	null	null	descrição	"Maltz (HUE) yellow"	null	null
30	Maltz.YR	Hue	é.pertencente.a	Sistema.Munsell	null	null	descrição	"Maltz (HUE) yellow-red"	null	null
31	Maltz.R	Hue	é.pertencente.a	Sistema.Munsell	null	null	descrição	"Maltz (HUE) red"	null	null
32	N10	Luminosidade	é.pertencente.a	Sistema.Munsell	null	null	descrição	"Nível de luminosidade 10 superior de valor branco"	null	null

Fonte: o autor

Os indivíduos representam conhecimentos factuais válidos, descrevendo aspectos associados a uma classe ou conceito. Por exemplo, todos os códigos da NBR 15.965 são considerados indivíduos

válidos que podem constar como declarações de uma ontologia escrita na sintaxe Manchester da seguinte forma:

Individual: bim:4A.01.01.04

Types: bim:Código.ABNT

Facts: bim:norma "NBR 15.965"

Facts: bim:parte "Parte 6"

Facts: bim:tema "Espaço funcional"

Facts: bim:descrição "Área privativa"

Facts: bim:subtema "Classificação de Áreas"

Facts:bim:categoria.revit "[OST_Rooms,OST_Areas,OST_MEPSpaces]"

Facts: bim:classe.ifc "IfcSpace"

Ou em sintaxe Turtle:

bim:4A.01.01.04 rdf:type owl:NamedIndividual , bim:Código ;

bim:norma "NBR 15.965" ;

bim:parte "Parte 6" ;

bim:tema "Espaço funcional" ;

bim:descrição "Área privativa" ;

bim:subtema "Classificação de Áreas" ;

bim:categoria.revit "[OST_Rooms,OST_Areas,OST_MEPSpaces]";

bim:classe.ifc "IfcSpace" .

Novos fatos (*Facts*) podem ser associados aos indivíduos e incorporados para manter, por exemplo, relações com os diversos esquemas de dados BIM existentes. Nesse caso, as propriedades *categoria.revit* e *classe.ifc* são as usadas como a ponte que permite relacionar o indivíduo declarado com um objeto específico da categoria do modelo de informação de Revit ou à classe do modelo de informação IFC. Ambos os modelos de dados certamente poderão utilizar esses valores como ponte ou gancho processual para auxiliar às APIs de aplicações computacionais a localizar, recuperar ou manipular um dado ou informação significativa.

7 INFRAESTRUTURA COGNITIVA PARA PROJETAR COM AGENTES DE IA

É possível aderir às dúvidas e às críticas que Landgrebe e Smith fazem sobre a Singularidade e a Inteligência Ampliada, que visionários como Vernor Vinge já antecipavam na década de 1990. No entanto, é mais difícil colocar em dúvida que a ampliação das fronteiras do conhecimento humano seja um fato em vias de ser concretizado. As possibilidades técnicas para que isso aconteça já estão dadas. O ato cognitivo humano de projetar, hoje em dia totalmente interpenetrado por processos digitais e algorítmicos, pode se beneficiar de ontologias que ampliem as possibilidades de integração consciente

dos fluxos de conhecimento produzidos pela inteligência projetual, difusamente presente nos objetos que nos rodeiam.

A tese que se defende neste escrito é que a problemática colocada pela IA se fundamenta muito menos na ideia de uma suposta ‘inteligência’ ou ‘pensamento maquinal’, e muito mais no papel que o conhecimento desempenha em contextos informatizados. A formulação de axiomáticas ontológicas — que representem o que se conhece factualmente sobre as coisas — deveria ser criada e publicada de modo aberto, permitindo um crescimento contínuo e ordenado de conhecimentos ao longo do processo projetual. Esse conhecimento estruturado será o que agentes humanos e maquinais poderão utilizar. Considera-se que a descrição factual de um edifício singular será construída mediante a adição sucessiva de conhecimentos, que passarão por diversos graus de completude ao longo de todo o processo. Os conhecimentos seriam somados e acumulados desde o momento da ideação, para posteriormente dar lugar aos processos de concepção; que, por sua vez, serão trabalhados com mais rigor e profundidade durante o desenvolvimento do projeto; momento em que será necessário considerar todos os aspectos legais e normativos; até, finalmente, chegar ao processo construtivo, já conhecido e, portanto, com maiores probabilidades de ser eficiente e ordenado. Todas as derivas posteriores relativas ao devir do objeto construído utilizarão a base de conhecimento que foi sendo formada ao longo de todo o processo. No entanto, a linearidade do processo descrito não significa que as soluções escolhidas sejam definitivas. Ao contrário, sabemos que os momentos de retorno e de desvio para outras soluções alternativas são uma realidade do processo projetual. É por isso que suspeitamos que a problemática apresentada pela IA dependa menos da ‘inteligência’ e mais do conhecimento acumulado. A hipótese do trabalho sustenta que *não pode existir inteligência sem conhecimento prévio*. Uma boa solução projetual pode ser consequência da sorte, do acaso ou da explosão combinatória de alternativas, mas não é isso o que nos interessa como projetistas conscientes. Nosso objetivo é criar condições que permitam a emergência de algo que chamamos de *inteligência projetual intencional assistida por meios computacionais*. A formulação de uma axiomática ontológica — capaz de expressar, computacional e humanamente, o conhecimento factual sobre as coisas — deveria permanecer aberta, permitindo um crescimento contínuo e ordenado do saber ao longo do processo projetual. A descrição factual de um edifício singular pode ser construída mediante a incorporação sucessiva de conhecimentos, avançando por diferentes graus de completude. As ontologias, entendidas como descrições da realidade projetual, constituiriam o lugar onde as diversas aplicações poderiam buscar seus dados para processamento. Os axiomas e asserções iniciais sobre o projeto podem ter origem em conhecimentos já sedimentados pela indústria em domínios disciplinares específicos. Aqui sugerimos que é estrategicamente necessário que as Normas Técnicas se transformem em uma dessas fontes originárias de conhecimento. Os processos heurísticos de inferência e descoberta de novos conhecimentos sobre instâncias particulares projetadas seriam

orientados por essas ontologias, que poderão ser reformuladas e ampliadas por novas ontologias que surgirão em paralelo aos projetos. O objetivo perseguido com o mecanismo construtor é permitir formular e alterar, de maneira ordenada, os sistemas de regras e axiomas declarados para cada projeto singular instanciado. O acoplamento com outras ontologias já consolidadas ajudaria a orientar o processo projetual, integrando a intencionalidade humana com mecanismos algorítmicos de raciocínio formal. Assume-se que o conhecimento ontológico de um objeto será sempre incompleto, porém expansível. Também se afirmou que as descrições ontológicas devem permitir a revisão sistemática de conhecimentos estabelecidos e ser integradas e endereçáveis como recursos RDF nos ambientes da Web Semântica. Esse é um passo que, no campo da construção, ainda é incipiente, quando comparado ao campo da biologia. Para que o complexo tecnológico BIM continue evoluindo, seria recomendável que as publicações atuais dos diversos sistemas normativos — que constituem bases de conhecimento estabelecidas pela experiência industrial — comecem a ser traduzidas para o formato OWL ou TTL e disponibilizadas de maneira irrestrita. As Normas Técnicas expressam o conhecimento industrial acumulado ao longo dos anos, mas ainda o fazem de modo textual, em arquivos pouco estruturados ou em silos de dados fechados. Portanto, parece lógico sustentar que ontologias de domínio técnico deveriam ser desenvolvidas e publicadas abertamente pelas organizações normativas, como a Associação Brasileira de Normas Técnicas. Manter uma política constante para criar e manter ontologias das normativas é uma necessidade aberta para o futuro projetual. Ontologias de produtos publicadas pelos fabricantes, que são os agentes que melhor conhecem seus produtos e processos, complementariam as tradicionais bibliotecas de componentes. Códigos municipais e exigências de segurança contra incêndio publicados pelo Corpo de Bombeiros poderiam se converter em ontologias, modificando o fluxo de validação e aprovação de projetos em um processo aberto e contínuo. Uma das dificuldades para implementar sistemas baseados em ontologias é a mudança de paradigma do modelo de informação utilizado, que deve estar baseado em RDF (Resource Description Framework), além do fato de que, para aproveitar plenamente a tecnologia, são necessários maiores esforços de articulação da infraestrutura de dados. As ontologias deveriam ser publicadas e expostas em ambientes de dados abertos e vinculados, possibilitando seu descobrimento e sua conexão com outros esquemas ontológicos, o que permitiria explorar todo o seu potencial. A publicação de grafos de conhecimento pode integrar o paradigma de sistemas de informação descentralizados em APIs ao paradigma de sistemas descentralizados em dados. O paradigma de dados estruturados como RDF depende de plataformas especializadas que permitam a ingestão de RDF, cujos conteúdos são conhecimentos declarados em sentenças de tríades (Sujeito, Predicado e Objeto). Essas plataformas permitem carregar arquivos OWL (*Ontology Web Language*) ou TTL (*Turtle*) e estão preparadas para aplicar consultas por meio de filtros SPARQL. Com esse objetivo, o construtor de ontologias programado para o Revit cria as ontologias em dois modos:

Ôntica geral: com classes, propriedades e indivíduos que são particulares conhecidos para todos os projetistas do domínio específico. Por exemplo, os códigos urbanísticos ou os códigos definidos pela ABNT são de fato particulares válidos para todos os projetistas.

Individual: bim:4A.01.01.04

Types: bim:Código

Facts: bim:norma "NBR 15.965"

Facts: bim:parte "Parte 6"

Facts: bim:tema "Espaço funcional"

Facts: bim:descrição "Área privativa"

Facts: bim:subtema "Classificação de Áreas"

Facts: bim:categoria.revit "[OST_Rooms , OST_Areas , OST_MEPSpaces]"

Facts: bim:classe.ifc "IfcSpace"

Modo projeto: extrai a ontologia a partir do modelo BIM de projetos reais, formada por indivíduos concretos de cada projeto, como no exemplo apresentado de um compartimento existente em um edifício institucional. Ou seja, algo que não existia, passou a ser um fato existente.

Individual: bim:Sala.D104

Types: bim:Ambiente

Facts: bim:é.dentro.de bim:Bloco.D

Facts: bim:nome.do.ambiente "Laboratório de Informática"

Facts: bim:código.do.ambiente "D104"

Facts: bim:latitude "-22.860190"

Facts: bim:longitude "-43.229833"

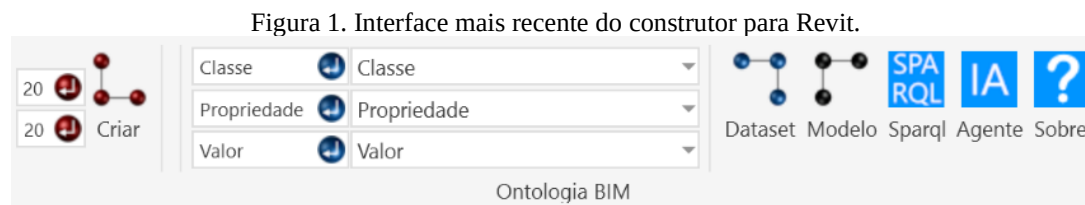
Facts: bim:altitude "5.60"

Facts: bim:unidade.acadêmica "POLI"

Facts: bim:descrição "LIG da POLI usado em aulas de Expressão Gráfica"

Facts: bim:categoria.revit "[OST_Rooms , OST_MEPSpaces]"

Facts: bim:classe.ifc "IfcSpace"



Fonte: o autor.

As ontologias hospedadas no repositório GitHub são atualizadas automaticamente por meio da API do repositório. Nas primeiras versões do construtor, trabalhou-se com a formalização da ontologia

OWL na sintaxe Manchester. Como foi mencionado, a preferência por essa sintaxe apoiava-se na recomendação de Gruber sobre a mínima codificação e na facilidade de leitura que oferece para agentes humanos. No entanto, existe um custo associado à incorporação das ontologias em plataformas especializadas. Em geral, essas plataformas preferem a ingestão de ontologias formatadas e escritas em linguagens de descrição RDF ou Turtle (TTL). O trabalho de tradução para TTL era realizado no programa Protégé, o que implicava a execução de um processo adicional. A atual versão cria também a ontologia neste formato.

Ontologias que permitem declarar predicados de conhecimento estabelecido (Xiao *et al.*, 2019) estão sendo integradas à arquitetura de sistemas de IA generativa modelados com técnicas conhecidas pelo acrônimo RAG (*Retrieval Augmented Generation*) (Zhaozhiming, 2024). Nesse campo, os desenvolvedores têm promovido um processo de convergência entre a IA baseada em sistemas vetoriais de NLP (*Natural Language Processing*), utilizados nos Grandes Modelos de Linguagem (LLMs), e a IA simbólica tradicional. A IA simbólica, formalizada com técnicas baseadas em ontologias, participa do processo fornecendo representações computacionais do conhecimento factual da realidade. Assim, o uso de agentes inteligentes com maior autonomia é resumido na denominação *Agentic IA*, e a convergência entre sistemas vetoriais e simbólicos é chamada de *IA neuro-simbólica* (Franz, 2025).

As Normas Técnicas expressam o conhecimento industrial acumulado ao longo dos anos, mas ainda o fazem de modo textual, em arquivos computacionalmente pouco estruturados, em silos de dados fechados ou, recentemente, por agentes inteligentes desarticulados. As ontologias em OWL de cada domínio técnico podem ser desenvolvidas e publicadas abertamente pelas organizações normativas, como a Associação Brasileira de Normas Técnicas (ABNT). A tese defendida no presente artigo é que a reconversão das normas, em forma e conteúdo, como assim também a estruturação política de uma infraestrutura de rede semântica nacional controlada, permitiria aplicar e explorar técnicas de pesquisa orientada a entidades para acessar as bases de conhecimento, ampliando o espectro de significados contextuais graças ao conjunto de relacionamentos semânticos definidos pelos atributos dos conceitos ontológicos (Devezas *et al.*, 2021). O presente trabalho é consonante com as pesquisas que sustentam que a lógica incorporada pelas ontologias de domínio associadas com agentes de LLM, ajudaria a configurar paulatinamente um ambiente melhorado e mais confiável de engenharia de prompts (Aasman, 2022) (Galkin, 2023) (Souza, *et al.*, 2024) (Zheng *et al.*, 2025).

Pode-se também intuir que a articulação entre sistemas simbólicos e ontológicos combinados com interfaces de voz, além de melhorar as consultas, possam também aumentar o componente puramente intelectual na tarefa de projetar, em detrimento do seu componente gestual. Em outras palavras, um potencial aumento da influência da mente sobre o corpo na tarefa projetual, auxiliando e complementando tecnologias computacionais em contextos assistivos. Embora seja tecnicamente

possível, acredita-se que seja uma decisão política a tarefa de montar e manter articulado um ecossistema que democratize o conhecimento industrial, permitindo que seja compartilhado e aberto. A tarefa, precisaria considerar as fontes de financiamento que permitam que a infraestrutura do sistema possa crescer e ser mantida de modo sustentável ao longo do tempo, uma vez que o devir é uma realidade do ambiente construído. A arquitetura de rede de conhecimentos normativos deveria ser complementada e integrada às ontologias de produtos, publicadas com autonomia pelos fabricantes que conhecem os seus objetos e processos, substituindo as tradicionais bibliotecas de componentes. Os códigos de Prefeituras e Corpo de Bombeiros associados à rede poderiam modificar os atuais fluxos de validação e aprovação de projetos por processos abertos e contínuos. Como benefício do trabalho de reconversão aponta-se um incremento da inteligência potencial incorporada aos sistemas de projeto, aumentando o grau de colaboração não só entre projetistas humanos, senão também entre humanos e máquinas ou entre máquinas, além de favorecer a abertura e manutenção de novas frentes de trabalho intelectual.

Em consonância com Landgrebe e Smith, destaca-se que é pertinente duvidar sobre a possibilidade de um acontecimento historicamente repentino como a Singularidade. Também pode-se duvidar de cenários em que surjam entidades ou seres de apresentem aptidões superiores como uma superinteligência pós-humana, como as visões de Vernor Vinge sugeriam. Entretanto, é mais difícil colocar em dúvida que graças ao desenvolvimento e uso intensivo de redes de conhecimento significativo e semanticamente estruturado, a expansão das fronteiras do conhecimento da humanidade seja de fato um horizonte possível e, o ato cognitivo orientado ao projeto, venha a se desenvolver em ambientes digitais abertos e híbridos dentro de uma infraestrutura que combine agentes humanos e artificiais que ainda está em formação. Fomentar uma política de desenvolvimento de ontologias e de manutenção da infraestrutura de servidores semânticos, pode ser um fator de aderência necessário entre agentes humanos e agentes algorítmicos cada dia mais autônomos, ordenando os fluxos de conhecimento factual produzidos por uma inteligência projetual que tenha a sua origem na consciência humana.

REFERÊNCIAS

- AASMAN, J. *The foundation of data fabrics and AI: Semantic Knowledge Graph*. Disponível em: <https://www.datasciencecentral.com/the-foundation-of-data-fabrics-and-ai-semantic-knowledge-graphs/> May, 2022. Acesso em: abril de 2025.
- AL-HAKAM, H.; SCHERER, R. J. *Integration of BIM-related bridge information in an ontological knowledgebase*. Conference: LDAC 2020 - 8th Linked Data in Architecture and Construction Workshop. 2020.
- AISH, R. *First Build Your Tools*. Em: Inside SmartGeometry: Expanding the Architectural Possibilities of Computational Design. pp.36-49. Londres: Wiley. 2013.
- BANG, Y.; CAHYAWIJAYA, S.; LEE, N.; DAI, W.; SU, D; WILIE, B.; LOVENIA, H.; JI, Z.; YU, T.; CHUNG, W.; DO, Q. V.; XU, Y; FUNG, P. *A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity*. ArXiv:2302.04023. 2023.
- BIM4REN. D4.2 BIM4REN repository: software and user guide. Disponível em: <https://bim4ren.eu/download/d4-2-bim4ren-repository-software-and-user-guide/> Acesso em: março de 2024
- BORGO, S.; GALTON, A.; KUTZ, O. *Foundational ontologies in action. understanding foundational ontology through examples*. Applied ontology, v.17. n°1. pp.1–16. 2022. DOI: 10.3233/AO-220265
- BEACH, T. H.; REZGUI, Y. *Semantic Encoding of Construction Regulations*. Proceedings of the 6th Linked Data in Architecture and Construction Workshop Londres, United Kingdom. 2018.
- BEETZ, J.; LEEUWEN, J. V.; VRIES, B. IfcOWL: *A case of transforming EXPRESS schemas into ontologies*. Artificial Intelligence for Engineering Design, Analysis and Manufacturing. 2009. v.23, n.1, pp.89-101. DOI: <https://doi.org/10.1017/S0890060409000122>
- BERNERS-LEE, T. *Linked Data*. 2009. Disponível em: <https://www.w3.org/DesignIssues/LinkedData.html> Acesso em: jul 2023.
- BERNERS-LEE, T.; FIELDING, R.; MASINTER, L. RFC 3986: *Uniform Resource Identifier (URI): Generic Syntax*. IETF Internet Engineering Task Force. 2005. Disponível em: <http://www.ietf.org/rfc/rfc3986.txt> Acesso em: jul 2023.
- BFO. *Basic Formal Ontology*. 2020. Disponível em: <https://basic-formal-ontology.org>
- BONDUEL, M.; ORASKARI, J.; PAUWELS, P.; VERFAUWEN, M.; KLEIN, R. *The IFC to Linked Building Data Converter-Current Status*. Em: Linked Data in Architecture and Construction Workshop 6. Londres. Proceedings [...]. Londres: CEUR-WS. 2018. v.2159. pp.34-43. Acesso em: mai 25, 2019. Disponível em: <http://ceur-ws.org/Vol-2159/>
- BRYANT, L. R. El principio óntico: boceto de una ontología orientada a objetos. FLORES PEÑA, G. R. (Tradução). Devenires , xviii , 35 (2017): 229-264. 2017.
- bSDD. *buildingSMART Data Dictionary*. (2025). Disponível em: <https://search.bsdd.buildingsmart.org/> Acesso em: mar 2025.

BUS, N.; ROXIN, A.; PICINBONO, G.; FAHAD, M. *Towards French Smart Building Code: Compliance Checking Based on Semantic Rules* Nicolas. Proceedings of the 6th Linked Data in Architecture and Construction Workshop. Londres, United Kingdom. 2018.

DEVEZAS, J.; NUNES, S. *A Review of Graph-Based Models for Entity-Oriented Search*. SN Computer Science. 2021. v.2, n.437 <https://doi.org/10.1007/s42979-021-00828-w>

ENGEL, H.; RAPSON, R. Tragsysteme. Hatje Cantz Verlag. 2007.

FIORAVANTI, A.; TRENTO, A. *Close Future: Co-Design Assistant How Proactive design paradigm can help*. Em: Proceedings of 37 eCAADe and XXIII SIGraDi Joint Conference, Architecture in the Age of the 4Th Industrial Revolution. SOUSA, J. P.; HENRIQUES, G. C.; XAVIER, J. P. (Eds.). São Paulo: Blucher. pp.155-162. DOI: 10.5151/proceedings-ecaadesigradi2019_516. 2019.

FLACH, P. Simply Logical. Intelligent Reasoning by Example. John Wiley, 1994.

FRANZ INC. (2025) *Agentic AI with AllegroGraph's Neuro-Symbolic Knowledge Graphs*. White paper. In. *Database. Trends and Application*. March, 2025. Disponível em: <https://allegrograph.com/wp-content/uploads/2025/03/Powering-GenAI-Apps-with-Knowledge-Graphs-3-2025-2.pdf>

GALKIN, M. *Neural Graph Databases. A new milestone in graph data management*. Disponível em: Towards Data Science. <https://towardsdatascience.com/neural-graph-databases-cc35c9e1d04f>, 2023. Acesso em: jan 2025.

GRUBER, T. R. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*. v.43. pp.907-928. 1995. DOI: <https://doi.org/10.1006/ijhc.1995.1081>

HORRIDGE, M.; KNUBLAUCH, H.; RECTOR, A.; STEVENS, R.; WROE, C. *A Practical Guide to building OWL ontologies using the Protégé-OWL plugin and CO-ODE Tools*. Ed.1.0. The University of Manchester. 2004.

ISOTANI, S.; BITTENCOURT, I. I. Dados abertos conectados. São Paulo, Novatec Editora, 2015.

KASSEL, G. *A plea for epistemic ontologies*. *Applied Ontology*. v.18. n.4. pp.367 397. Dezembro 2023. DOI: 10.3233/AO-230031

LANDGREBE, J.; SMITH, B. *Why Machines Will Never Rule the World*. Artificial Intelligence without Fear. New York. Routledge. 2023.

LASKAR, T. R.; BARI, M. S.; RAHMAN, M.; BHUIYAN, A. H.; JOTY, S.; HUANG, J. *A systematic study and comprehensive evaluation of ChatGPT on benchmark datasets*. Em: Findings of the Association for Computational Linguistics: ACL 2023, pp. 431–469, Toronto, Canada. Association for Computational Linguistics. 2023. DOI: <https://doi.org/10.48550/arXiv.2305.18486>

LEVESQUE, H. J. *Foundations of a functional approach to knowledge representation*. *Artificial Intelligence*, v.23, pp.155-212. 1984.

MACIEL LIMA, J. P.; REIS PEREIRA, C. O princípio de não contradição: princípio ontológico, ôntico e gnosiológico do ser. The principle of non-contradiction: ontological, ontic and gnosiological principle of being. *Revista Opinião Filosófica*, v. 15, n. 2, e1078, p. 1-13, julho-dezembro, 2024 - ISSN: 2178-1176.

MENEGOTTO, J. L. O modelo digital. Técnica e arte algorítmica em BIM. Ed. Interciencia. Rio de Janeiro, 2023.

_____. *Ontologia BIM. Alguns Aspectos do Conhecimento Projetual: o Prédio, as IFC e as OST*. Sigradi 2023. Accelerated Landscapes | Centro Universitario Regional Este (CURE) Facultad de Arquitectura, Diseño y Urbanismo | Universidad de la República. 2023.

_____. *The neuro symbolic AI: Formalize ontologies of BIM product catalogs*. Sigradi 2025. Meta - Responsive Approaches faud. UNC. 2025.

MENGTIAN, Y.; LLELEWELLYN, T.; WEBSTER, C.; SHEN, X.; XIONGYI, L.; HUAQUAN, Y. *An ontology-aided, natural language-based approach for multi-constraint BIM model querying*. Journal of Building Engineering. n.76. Elsevier, 2023.

NEUHAUS, F. *Ontologies in the era of large language models - a perspective*. Applied Ontology. v.18. n.4. pp.399-407. Dezembro 2023. DOI: 10.3233/AO-230072

NEUHAUS, F.; GRENON, P.; SMITH, B. *A Formal Theory of Substances, Qualities, and Universals*. Em VARZI, A.; VIEU, L. (eds.), Proceedings of FOIS 2004. International Conference on Formal Ontology and Information Systems, Turin, 4-6. November 2004.

POVEDA-VILLALÓN, M.; GARCIA-CASTRO, R. Extending the SAREF ontology for building devices and topology. Proceedings of the 6th Linked Data in Architecture and Construction Workshop. Londres, United Kingdom. 2018.

RASMUNSEN, M. H.; HVIID, C. A.; KARLSHOJ, J. *Web-based topology queries on a BIM model*. Em: Linked Data in Architecture and Construction Workshop, 5. Proceedings [...]. Dijon. 2017.

SCHULZL, O.; ORASKARY, J.; BEETZ, J. *Lessons Learned from Designing and Using bcfOWL*. Proceedings LDAC2023 – 11th Linked Data in Architecture and Construction, jun. 15–16, Matera, Italy. 2023.

SMITH, B. *New desiderata for biomedical terminologies*. Em: MUNN, K.; SMITH, B. (eds.). Applied Ontology: An Introduction, Frankfurt-Lancaster: Ontos. pp.83-109. 2008.

SMITH, B. *Beyond Concepts: Ontology as Reality Representation*. Em VARZI, A.; VIEU, L. (eds.), Proceedings of FOIS 2004. International Conference on Formal Ontology and Information Systems, Turin, 4-6. November 2004.

SOUZA, D. L. de; RUSCHEL, R. C. Análise de ontologias para construção civil utilizando ferramentas automáticas baseadas em métricas de qualidade. PARC: Pesquisa em Arquitetura e Construção, Campinas, SP, v. 15, n. 00, p. e024012, 2024. DOI: 10.20396/parc.v15i00.8673829.

USCHOLD, M.; GRUNINGER, M. Ontologies: Principles, methods and applications. Em: The Knowledge Engineering Review. v.11, n.2. 1996.

VINGE, V. *The Coming Technological Singularity: How to Survive in the Post-Human Era*. Em: Vision-21: Interdisciplinary Science and Engineering in the Era of Cyberspace. LANDIS, G. A. (ed), NASA Publication CP-10129. pp.11-22. 1993.

XIAO G., DING L., COGREL B., & CALVANESE D. *Virtual knowledge graphs: An overview of systems and use cases*. (“Accessing scientific data through knowledge graphs with Ontop - Cell

Press”) Data Intelligence 1(2019), 201-223. doi: 10.1162/dint_a_00011 Received: February 24, 2019; Revised: April 7, 2019; Accepted: April 26, 2019

YOUNG R. I. M.; GUNENDRAN A. G.; CUTTING-DECELLE A. F.; GRUNINGER M. *Manufacturing knowledge sharing in PLM: a progression towards the use of heavy weight ontologies*. International Journal of Production Research. v.45. n.7. pp.1505–1519. Abril 2007.

ZHAOZHIMING. *Advanced RAG Retrieval Strategies Using Knowledge Graphs*. In Generative IA. Disponível em: <https://medium.com/generative-ai/advanced-rag-retrieval-strategies-using-knowledge-graphs-12c9ce54d2da>. Jun, 2024.

ZHENG, C.; WONG, S.; SU, X.; TANG, Y.; NAWAZ, A.; KASSEM, M. *Automating construction contract review using knowledge graph-enhanced large language models*. Em. Automation in Construction, Vol. 175, 2025. <https://doi.org/10.1016/j.autcon.2025.106179>.