Validação de uma Metodologia Ontológica no Domínio do Mercado Imobiliário

Matheus da Cruz Percine Pinto

[1] Universidade Federal do Rio de Janeiro, Brazil
[2] matheuspercinep@gmail.com
[3] matheuscpp@dcc.ufrj.br
https://github.com/MatheusPercine

Abstract. O mercado imobiliário produz um grande volume de dados heterogêneos, cuja análise é dificultada pela ausência de padronização terminológica e interoperabilidade semântica. Essa limitação compromete a automação de buscas complexas e a descoberta eficiente de valor. Este trabalho propõe uma solução baseada na Web Semântica, aplicando e validando a robustez de uma metodologia de referência para desenvolver uma nova ontologia voltada ao mercado de aluguel de imóveis. A ontologia foi modelada para representar conceitos como Apartamento, Comodidades hierarquizadas, Atores e Transações. Um script em Python foi utilizado para populá-la com dados de dois conjuntos públicos, resultando em um grafo de conhecimento. A validação ocorreu por meio de consultas SPARQL, testando a capacidade do modelo de resolver sinônimos e inferir classificações (e.g., apartamentos que aceitam pets). Os resultados indicaram que o modelo integrou os dados com sucesso e respondeu corretamente a consultas por categorias abstratas (e.g., comodidades de lazer). Conclui-se que o modelo ontológico desenvolvido é robusto, e a metodologia adotada mostra-se aplicável a contextos mais amplos do setor imobiliário, servindo como base para futuras aplicações de análise inteligente.

Keywords: Ontologia · Web Semântica · Mercado Imobiliário · Interoperabilidade de Dados · Grafo de Conhecimento.

1 Introdução

O link para o projeto se encontra em: https://github.com/MatheusPercine/Ontology-For-Real-Estate-Market

1.1 Falta de Padronização e Interoperabilidade dos Dados

Três anúncios exemplificam o problema terminológico: "gym", "fitness center" e "academia" referem-se a espaços de exercícios, mas sistemas computacionais os tratam como termos distintos. Essa ausência de vocabulário comum compromete a interoperabilidade, ou seja, a capacidade de diferentes sistemas trocarem

informações com significado compartilhado. Por exemplo, consultas por "área de lazer externa" não retornam resultados com termos como "pátio" ou "varanda", devido à falta de equivalência reconhecida.

Essa situação obriga a normalização manual dos dados, processo lento, custoso e sujeito a erros, deixando os dados fragmentados e dificultando análises, recomendações e ferramentas para consumidores e empresas.

1.2 Uso de Ontologia

A Web Semântica oferece uma solução eficaz: ontologias, que funcionam como mapas conceituais que definem termos e relações de modo compreensível para máquinas, permitindo interpretação lógica dos dados.

Por exemplo, a regra "Se um apartamento permite gatos ou cães, então é PetFriendlyApartment" permite inferir informações não explícitas, como listar imóveis que aceitam animais. Assim, ontologias estruturam e geram conhecimento, padronizando vocabulário, hierarquias e regras, o que viabiliza a interoperabilidade entre sistemas.

1.3 Trabalho Relacionado

Este projeto baseia-se no artigo "Towards a unified ontology for monitoring ecosystem services", que enfrentou problema similar em ciências ambientais: inconsistência terminológica dificultando análises globais.

A solução foi a ESM Ontology, mapa conceitual padronizado que define conceitos como Ecossistema e Bem Ecossistêmico, permitindo organização e interpretação dos dados por humanos e máquinas.

A validação usou dados reais de estudos no Canadá e “perguntas de competência” que comprovaram a precisão do modelo. O destaque do artigo está no método rigoroso para construir e validar ontologias em domínios complexos.

1.4 Objetivo do Projeto

Embora a metodologia tenha sido validada no contexto ambiental, este projeto visa testá-la em um domínio distinto: o mercado imobiliário. O foco é a generalidade e robustez do método.

Para isso, três etapas foram seguidas: criação da ontologia imobiliária, população com dados reais e execução de consultas SPARQL para validar o modelo, analisando sua capacidade de trazer inteligência e padronização ao domínio.

2 Criação da Ontologia

Para o desenvolvimento da ontologia, foi utilizada a ferramenta Protégé, um ambiente integrado voltado à engenharia de ontologias, com suporte aos padrões RDF, RDFS e OWL 2. Sua interface gráfica intuitiva facilita a modelagem visual de classes e propriedades.

Optou-se por nomear classes e propriedades em inglês, visando interoperabilidade global. A maioria dos vocabulários, padrões do W3C e ontologias de referência no contexto da Web Semântica está definida nesse idioma. Assim, adotar o inglês como padrão estrutural garante compatibilidade com esse ecossistema, favorecendo integrações futuras e a reutilização do modelo por comunidades internacionais.

Arquivo da ontologia se encontra em: apartments.ttl

2.1 Classes

O conceito central do modelo é o Apartamento, entidade a ser descrita e analisada. Para compreendê-lo integralmente, a ontologia o conecta a elementos fundamentais como localização, características físicas e os envolvidos em sua negociação.

A primeira relação é com a Localização. Em vez de tratá-la como texto simples, a ontologia define um conceito próprio com atributos como cidade, estado e coordenadas geográficas, permitindo a compreensão do contexto espacial. O apartamento também possui atributos diretos e mensuráveis, como número de quartos, banheiros e valor do aluguel, considerados características intrínsecas da unidade.

Foi estabelecida uma distinção entre a estrutura física e os itens adicionais. Para isso, o modelo define duas categorias principais: Espaços Internos e Comodidades. O apartamento "possui espaços", como Cozinha, Sala de Estar ou Quintal, que integram sua estrutura, e "possui comodidades", como Academia, Piscina ou Porteiro, representando elementos adicionais.

Esses conceitos foram organizados em hierarquias. Dentro da categoria Comodidade, por exemplo, há subcategorias como Comodidade de Lazer e de Segurança. Assim, o sistema reconhece que Academia e Piscina são comodidades de lazer, enquanto Porteiro e Alarme são de segurança. Essa estrutura hierárquica permite buscas abstratas e a interpretação semântica dos termos, enfrentando a falta de padronização.

Para representar o contexto comercial, a ontologia inclui os conceitos de Ator e Transação. Atores são pessoas ou empresas envolvidas, como Inquilino e Proprietário. A Transação representa o evento do aluguel, estando ligada a um apartamento e a seus participantes. Essa estrutura permite não apenas descrever o imóvel, mas também registrar e analisar suas interações comerciais.

2.2 Propriedades de Objeto

No “mapa de conceitos”, as conexões entre conceitos são chamadas de Propriedades de Objeto, funcionando como verbos que ligam sujeito e objeto.

A relação entre Apartamento e sua Localização é estabelecida pela propriedade hasLocation, que associa o imóvel a dados geográficos como cidade, estado, latitude e longitude. A propriedade hasSpace indica que um apartamento é formado por diversos Espaços Internos, como cozinha ou sala de estar.

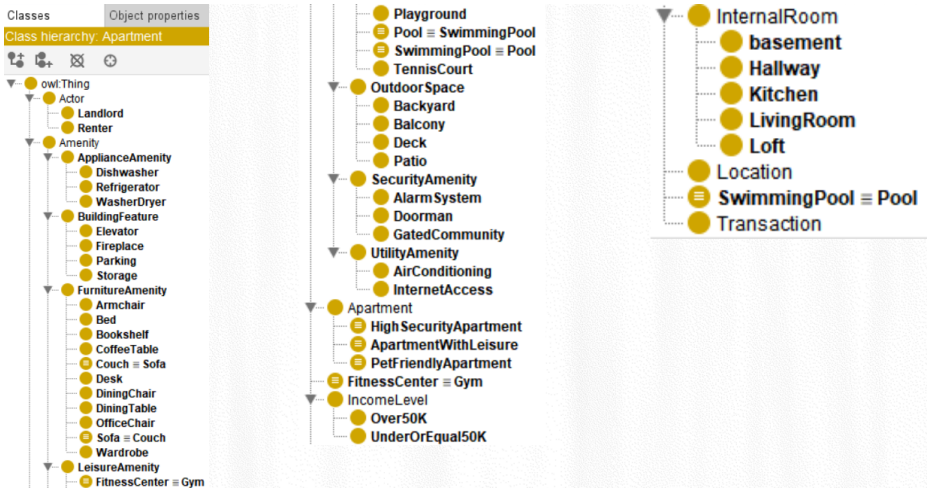

Fig. 1. Classes da ontologia.

Características e serviços adicionais são representados pela propriedade hasAmenity, que conecta o apartamento a comodidades, como academia, permitindo buscas por categorias abstratas, como "comodidades de lazer".

Para modelar transações econômicas, a propriedade concernsApartment vincula toda Transação a um Apartamento específico. As propriedades hasRenter e hasLandlord ligam a Transação aos Atores (inquilino e proprietário), definindo seus papéis no negócio. Essas propriedades ampliam a ontologia para descrever interações comerciais e seus participantes.

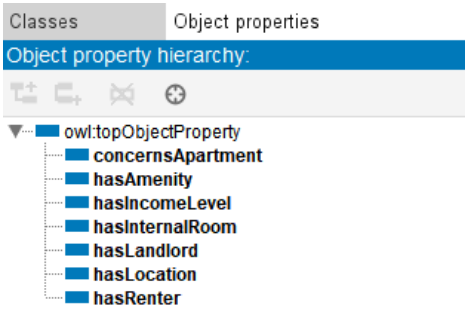

Fig. 2. Propriedades de Objeto

2.3 Propriedades de Dado

Uma ontologia armazena fatos e atributos específicos de cada item por meio das Propriedades de Dado, que ligam conceitos a valores literais, como números, textos ou booleanos.

A propriedade hasPrice conecta um apartamento ao seu preço de aluguel (valor decimal). Propriedades como hasBedrooms e hasBathrooms indicam a quantidade de quartos e banheiros. Propriedades booleanas, como allowsCats e allowsDogs, indicam se pets são permitidos.

Modelar "quartos" e "banheiros" como Propriedades de Dado, em contraste com "cozinha" como Classe, é estratégico. Quantidades são atributos diretos, enquanto espaços como cozinha, com tipos e características (ex.: americana, gourmet), demandam um conceito mais detalhado (Classe). Essa abordagem híbrida mantém o modelo simples onde possível e rico onde necessário.

A classe Location usa propriedades como hasCityName (texto) e hasLatitude e hasLongitude (decimais) para armazenar informações geográficas.

A classe Actor, que representa pessoas, inclui propriedades como hasAge (inteiro) e hasProfession (texto), além de ligação a categorias de renda por propriedade de objeto.

Na hierarquia de comodidades, a subclasse Parking possui a propriedade numberOfSpaces para vagas, e a subclasse Pool usa isHeated para indicar se é aquecida. Isso permite descrever características específicas, tornando o modelo detalhado, flexível e apto a responder perguntas precisas sobre o mercado imobiliário.

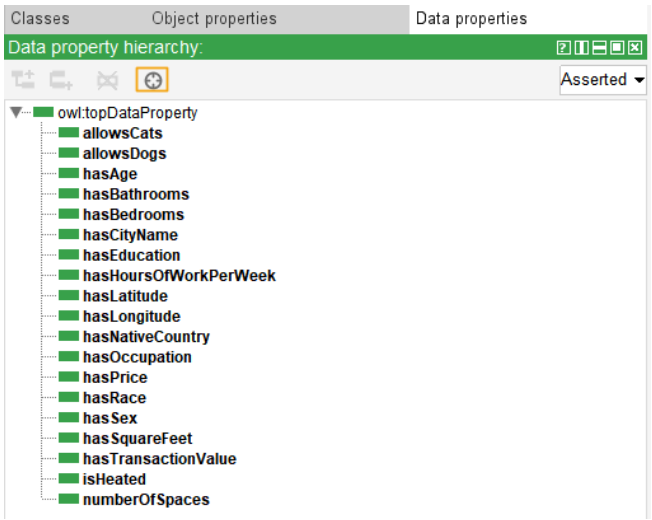

Fig. 3. Propriedades de dado.

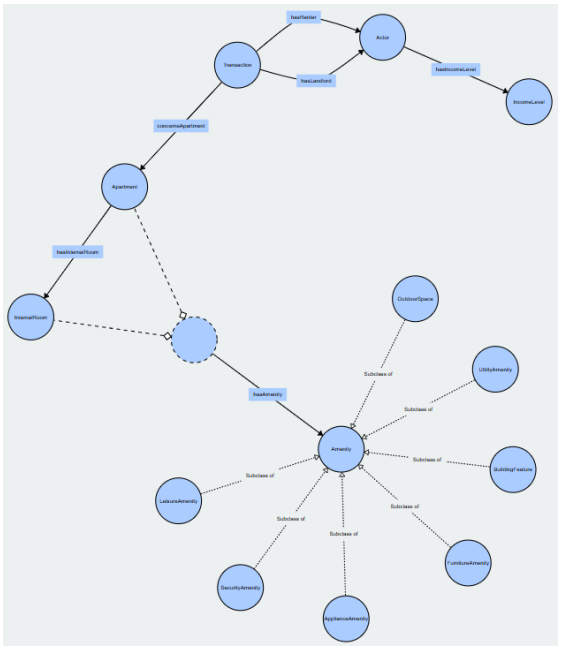

Fig. 4. Grafo de Conhecimento gerado pela ferramenta WebVOWL. A omissão de informações possibilita melhor entendimento.

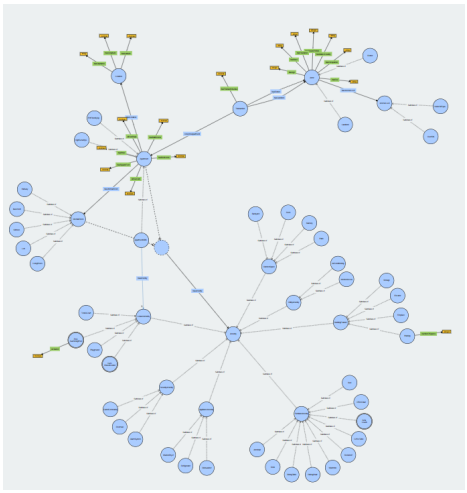

Fig. 5. Grafo de Conhecimento Completo pela ferramenta WebVOWL.

3 Os Datasets

Foram usados dois datasets: um relativos à informações de apartamentos para aluguel nos EUA e outro relativos à indivíduos dos EUA. Esses datasets se relacionam com as classes Apartment e Actor da ontologia.

4 Apartment Rent Data

A base selecionada contém dados detalhados de aluguéis de apartamentos, com 99.492 instâncias, sem explicitar origem ou método de coleta, apenas indicando licença MIT.

Arquivo: apartments_for_rent_classified_100K.csv

4.1 Conteúdo

Cada instância possui os seguintes atributos:

- ID: identificador único do anúncio
- Category: categoria do anúncio
- Title: título do anúncio
- Body: corpo do anúncio
- amenities: comodidades incluídas no aluguel
- bathrooms: número de banheiros
- bedrooms: número de quartos
- Currency: moeda de pagamento
- Fee: presença de taxa no aluguel
- Has_photo: indica foto no anúncio
- Pets_allowed: permissão para animais
- Price: preço do aluguel
- Price_display: preço exibido no anúncio
- Square_feet: tamanho em metros quadrados
- Address: endereço
- CityName: cidade
- State: estado
- Latitude: latitude
- Longitude: longitude
- Source: fonte do anúncio
- Time: tempo desde a criação do anúncio

4.2 Pré-processamento

Foi realizado pré-processamento dos dados brutos com Python e bibliotecas de análise para normalizar informações essenciais. A coluna amenities, originalmente texto livre, foi transformada em múltiplas colunas binárias indicando presença/ausência de comodidades, facilitando o mapeamento às classes correspondentes.

A coluna pets_allowed foi dividida em duas booleanas, cats e dogs, para detalhamento. Também foram removidos outliers que comprometiam a média dos preços.

Arquivo: Preprocessamento.ipynb

4.3 Retirada de Atributos Não Relevantes

Foram excluídos atributos que não agregavam informação útil ou apresentavam variância próxima de zero:

1. Sem informação relevante ou duplicados: ID, Body, Title, Price_display, Address, Source.
2. Variância quase nula: Category, Currency, Fee, Price_type.

4.4 Completude e Redundância

Tratou-se dados ausentes e instâncias duplicadas.

Dados Ausentes Para localização (latitude, longitude, estado, cidade) com valores nulos em poucas instâncias (302 para cidade/estado, 25 para coordenadas), optou-se pela exclusão, já que não havia substituição adequada.

Para bathrooms e bedrooms, valores ausentes foram assumidos como zero.

Instâncias Duplicadas Removidas 129 instâncias repetidas.

4.5 Outliers

Instâncias com price além de 3 desvios padrão da média ($z >= 3$) foram excluídas, totalizando 998 remoções.

4.6 Transformação dos Dados

Amenities e pets_allowed foram convertidos em variáveis dummy, criando 29 novas colunas (27 para amenities, 2 para pets). Ao final, o dataset contém 39 colunas e 97.935 instâncias.

5 Base: Adult Census Income

Arquivo: adult.csv

Para que a ontologia representasse interações reais, foram incluídos dados sobre as pessoas no mercado imobiliário. Utilizou-se a base pública "Adult Census Income", derivada do censo dos EUA de 1994, originalmente para prever se alguém ganha mais ou menos de 50 mil dólares anuais. Com mais de 32 mil indivíduos, o dataset é adequado para popular a classe Actor, adicionando uma

dimensão humana ao modelo e permitindo análises sobre possíveis inquilinos e proprietários.

Fornecido pelo U.S. Census Bureau, o conjunto traz atributos que permitem criar perfis socioeconômicos diversos. Embora a renda seja categórica (acima ou abaixo de 50 mil dólares), a informação é relevante para modelagem.

Diferentemente dos dados dos apartamentos, que demandaram pré-processamento, o dataset adult.csv foi utilizado em seu estado original, pois já passou por limpeza e curadoria, estando em formato coeso e sem erros evidentes.

Enquanto os dados dos imóveis contêm texto livre e desestruturado, o dataset adulto apresenta atributos atômicos como idade, sexo e horas semanais, que permitem mapeamento direto para as propriedades da classe Actor, sem transformações complexas.

Considerando o foco no mercado imobiliário e na validação da ontologia, optou-se por não realizar pré-processamento aprofundado nos dados dos atores, evitando análises estatísticas socioeconômicas que extrapolariam o escopo do projeto.

5.1 Conteúdo

Cada instância presente na base de dados possui os seguintes atributos:

- age: idade
- workclass: se atua no setor privado, governamental etc (ex: private)
- education: nível de graduação (ex: bachelors)
- education.num: número que representa o nível de graduação
- marital.status: solteiro, casado, divorciado, etc
- occupation: profissão
- relationship: Unmarried, Not-in-family, Husband, etc
- race: raça do indivíduo
- sex: sexo do indivíduo
- capital.gain: lucro obtido com a venda de ativos (ex: ações, imóveis), em um ano.
- capital.loss: perda financeira sofrida ao vender ativos por um valor menor do que o preço de compra, em um ano
- hours.per.week: horas de trabalho gastas pelo indivíduo na semana
- native.country: país no qual o indivíduo nasceu
- income: salário ($<= 50K$ ou $> 50K$)

6 População da Ontologia

Temos 2 arquivos para esta etapa: PopularOntologia.ipynb e PopularOntologia(CodExplicado).ipynb.

O segundo arquivo somente é uma versão bem mais comentada do primeiro arquivo, com o objetivo e documentar e explicar o processo.

O arquivo produzido é: knowledge_graph_final.ttl

6.1 Ferramenta

Com a ontologia estruturada, o projeto avançou para a fase de população do grafo de conhecimento, preenchendo o esquema com dados concretos.

Para essa tarefa automática e eficiente, foi usada a linguagem Python, que, por meio da biblioteca pandas, manipula dados, e com a rdflib cria e gerencia triplas RDF (Sujeito-Predicado-Objeto). Assim, Python traduz as informações das planilhas para o universo conectado e semântico do grafo.

6.2 O Processo

O script Python seguiu uma sequência lógica para garantir a integridade do modelo final. Primeiro, carrega a estrutura vazia da ontologia em um grafo na memória. Depois, abre os arquivos CSV com dados dos apartamentos e atores. Em seguida, mapeia e gera triplas: para cada linha, cria instâncias e conexões definidas na ontologia. Por exemplo, ao ler um apartamento com academia, cria uma instância da classe Gym e a conecta ao apartamento via hasAmenity, transformando um valor binário em uma relação semântica explícita.

Para validar a conexão entre domínios distintos, o script criou uma ligação entre instâncias de Apartment e Actor. Como os dados originais não possuíam essa relação, foi implementada uma simulação que, para cada apartamento, seleciona aleatoriamente um ator como inquilino em uma nova instância de RentalTransaction. Essa etapa visa gerar um grafo conectado para validar consultas semânticas complexas, não simular o mercado real.

Por fim, o script salva o conjunto de milhões de triplas em um único arquivo representando o grafo completo, pronto para análise.

7 Validação da Ontologia e Análise dos Resultados

Após criar o esquema ontológico e popular o grafo com dados dos dois datasets, a etapa final foi validar o modelo. O objetivo era verificar se a ontologia resolve a falta de padronização, permite interoperabilidade e viabiliza análises complexas inviáveis em bancos tradicionais. Para isso, utilizou-se o banco de grafos GraphDB, que processa regras lógicas definidas em OWL.

A validação ocorreu por meio de 15 Questões de Competência (CQs), organizadas em cinco categorias (C1 a C5), cada uma testando uma capacidade específica do grafo. As CQs foram traduzidas para SPARQL e executadas no grafo. Resultados e análises seguem.

As consultas SPARQL e respostas estão no arquivo CompetencyQuestions.txt.

7.1 C1: Validação de Dados e Filtros

Testou-se a correta população dos dados e a execução de consultas com múltiplos filtros sobre dados diversos. Perguntas como a contagem de apartamentos em Chicago que aceitam cães e têm mais de um banheiro retornaram 49 instâncias.

Também foram calculados preços médios e filtrados atores por idade e profissão. Os resultados confirmam o mapeamento correto e a capacidade do sistema em lidar com filtros numéricos, textuais e booleanos, estabelecendo base confiável para análises semânticas.

7.2 C2: Tratamento de Sinônimos

Validou-se o tratamento de sinônimos por meio da regra owl:equivalentClass, que unifica termos com mesmo conceito. Consultas para "Gym" e "FitnessCenter" retornaram o mesmo número (36.860), e para "Pool" e "SwimmingPool" retornaram 42.970 apartamentos. Isso comprova que o sistema reconhece equivalência semântica, resolvendo ambiguidades terminológicas.

7.3 C3: Consulta por Categoria Abstrata

Testou-se a navegação hierárquica usando rdfs:subClassOf, permitindo buscas por categorias gerais como "comodidades de lazer". Consultas indicaram 33 apartamentos com alguma LeisureAmenity em Nova York e 8.933 com SecurityAmenity. Um teste para preço médio de apartamentos com mobília retornou zero, evidenciando ausência desse dado na fonte, não falha do modelo.

7.4 C4: Inferência de Regras Complexas

Testou-se a geração de conhecimento novo por regras lógicas complexas, aplicando condições de interseção (owl:intersectionOf). Foram identificados apartamentos HighSecurityApartment (com porteiro e portão) e de "alto padrão" (acima de 3500 dólares com piscina e porteiro). Resultados positivos mostram que o motor de inferência processa axiomas e classifica corretamente as instâncias.

7.5 C5: Análise Cruzada de Domínios

Testou-se a interoperabilidade final, cruzando dados dos apartamentos com dados demográficos via instâncias de transação simuladas. Uma consulta indicou "Craft-repair" como profissão mais comum entre inquilinos de imóveis "pet-friendly", refletindo a distribuição demográfica. Outra comparação de preços médios entre atores de alta e baixa renda mostrou valores similares (1470 dólares), esperado pela atribuição aleatória. Esses resultados confirmam a capacidade do modelo em criar um grafo unificado para análises integradas.

8 Conclusão e Trabalhos Futuros

Este trabalho validou a robustez de uma metodologia ontológica para resolver problemas de padronização e interoperabilidade no complexo mercado imobiliário. A validação, por meio de rigorosas Questões de Competência, mostrou

que o grafo resultante realiza filtros básicos, resolve ambiguidades de sinônimos, permite consultas por categorias abstratas via hierarquias e infere novo conhecimento por regras lógicas. A principal contribuição foi demonstrar a integração e análise cruzada entre dois datasets distintos, confirmando a interoperabilidade semântica e a generalização da metodologia.

Apesar do sucesso, a abordagem apresenta limitações. O tratamento manual de sinônimos por axiomas de equivalência é eficaz para prova de conceito, mas pouco escalável. Para aplicações em escala, recomenda-se a integração com léxicos externos, como WordNet, para ampliar a cobertura automática de sinônimos. Além disso, o modelo pode ser enriquecido com dimensão temporal para analisar evolução de preços e com mais características possíveis de apartamentos e pessoas. Mostramos que a estrutura ontológica foi validada e as consultas SPARQL estão sintaticamente corretas, ou seja, o mecanismo funciona, porém ainda não confirmamos se as informações retornadas estão corretas, portanto cabe a criação de um gabarito para validar a acurácia das informações recuperadas. Diante dessas limitações, há espaço para trabalhos futuros que as mitiguem.

References

1. Shashank, S.: Apartments for Rent - Classified Dataset. Kaggle. `https://www.kaggle.com/datasets/shashanks1202/apartment-rent-data/data?select=apartments_for_rent_classified_100K`, last accessed 2025/07/03
2. Dua, D., Graff, C.: UCI Adult Census Income Dataset. Kaggle. `https://www.kaggle.com/datasets/uciml/adult-census-income`, last accessed 2025/07/03
3. F. Affinito, J.M. Holzer, M.-J. Fortin, A. Gonzalez: Towards a unified ontology for monitoring ecosystem services **6**, 100217 (2025). `https://www.sciencedirect.com/science/article/pii/S2212041625000300`, last accessed 2025/07/03
4. Amazon Web Services (AWS): O que é interoperabilidade? `https://aws.amazon.com/pt/what-is/interoperability/`, last accessed 2025/07/03