# Garantindo a Qualidade de Dados na Fusão de Dados Conectados: Um caso de uso de *SHACL* em dados abertos de Mobilidade e Educação de Curitiba

**Otávio Thomas Bertucini[1], Rita C. G. Berardi[1], Mateus G. Belizario[1], Nadia Kozievitch[1]**

[1]Universidade Tecnológica Federal do Paraná o Sul (UTFPR)
Av. Sete de Setembro 3165, Rebouças, 80230-901 Curitiba – PR – Brasil

otaviobertucini@gmail.com, {ritaberardi, nadiap}@utfpr.edu.br

***Abstract.*** *Smart cities are a context which can gain great advantage in the format and growth of data in the semantic web, as volume and connection increase the quality of data analysis. However, the quantitative growth of data must happen with quality assurance. This work aims to verify the quality of data in the fusion of connected data, through the dimensions of quality accuracy, consistency and conciseness. For the quality constraints to be verified, the SHACL language (Shapes Constraint Language) was used, and a Python script was created to perform the verification. The tests were performed on a set of connected open data from the domain of urban mobility and education in the city of Curitiba.*

***Resumo.*** *As cidades inteligentes são um contexto que pode obter grande vantagem no formato e no crescimento de dados na web semântica, visto que o volume e a conexão aumentam a qualidade das análises de dados. No entanto, o crescimento quantitativo de dados deve acontecer com garantia de qualidade. Este trabalho tem como objetivo a verificação de qualidade de dados na fusão de dados conectados, por meio das dimensões de qualidade acurácia, consistência e concisão. Para especificar as restrições de qualidade a serem verificadas foi utilizada a linguagem SHACL (Shapes Constraint Language) e para a execução da verificação foi criado um script em Python. Os testes foram realizados em um conjunto de dados abertos conectados do domínio de mobilidade urbana e educação na cidade de Curitiba.*

## 1. Introdução

A Web Semântica tem o potencial de revolucionar a maneira como descobrimos, acessamos, integramos e usamos dados, pois permite que dados de contexto diferentes se conectem [HEATH; BIZER, 2011] e dessa forma aumentar a flexibilidade, adaptabilidade e eficiência da gestão da informação no setor público e privado. No contexto de cidades inteligentes, por exemplo, é necessário integrar dados de fontes diferentes, com características semânticas diferentes. A criação e aplicação de um modelo de informação unificado permite obter uma visão mais completa da atividade urbana, integrando todas as fontes de dados em um único lugar [NAPHADE et al., 2011]. Um fator importante para a qualidade das inferências feitas a partir de dados, sejam eles da Web Semântica ou de modelos tradicionais, é a quantidade de dados disponível para análise [HALEVY; NORVIG; PEREIRA, 2009]. Dessa forma é

desejável que modelos de dados cresçam cada vez mais e se tornem cada vez mais favoráveis para a tomada de decisões. Nos últimos anos, houve um grande crescimento no número de informações geradas e armazenadas na Web. Em 2015 foram identificadas mais de 37 bilhões de triplas provenientes de mais de 650 mil documentos com dados acessíveis ao público [RIETVELD; BEEK; SCHLOBACH, 2015]. Além do mais, houve um aumento no interesse do setor privado pelo uso das tecnologias e metodologias da Web Semântica por empresas como a CNN e  o Facebook, demonstrando que existe uma tendência de mais dados serem publicados e utilizados na rede de dados.

Apesar de um grande volume de dados ser de interesse para vários domínios de aplicação, principalmente cidades inteligentes, esse crescimento precisa acontecer com um controle de qualidade. Além do mais, com o advento do acesso aos dados abertos, as plataformas precisam estar preparadas para uma evolução natural da quantidade de dados, mas também com sua qualidade, tendo em vista que a adição de novos dados pode ferir a consistência, precisão e concisão em um conjunto de dados. A cidade é uma grande geradora de dados e a sua atualização e crescimento é inerente aos dados. Portanto é fundamental verificar se a fusão dos novos dados com os dados originais não resultam em um conjunto final com baixa qualidade e consequentemente inútil para a inferência de conhecimento.

Este trabalho tem como objetivo a verificação de qualidade de dados na fusão de dados conectados, considerando as dimensões de qualidade acurácia, consistência e concisão. Para isso, o mecanismo de verificação foi construído na linguagem SHACL[1] (*Shapes Constraint Language*) e testado em um conjunto de dados abertos conectados do domínio de mobilidade urbana e educação na cidade de Curitiba. Também foi criado um script em Python que permite a execução deste mecanismo em dois conjuntos de dados no formato RDF[2] (*Resource Description Framework*).

## 2. Trabalhos Relacionados

Paulheim e Stuckenschmidt (2016) mostraram que é possível criar algoritmos de inferência que avaliam a consistência de um determinado conjunto de dados semânticos através do uso de aprendizado de máquina. Os autores explicam que o algoritmo de inferência utilizado foi implementado através da biblioteca pyshacl da linguagem Python e foi utilizado para realizar as validações com os shapes em SHACL. Spahiu, Maurino e Palmonari (2018) propuseram uma metodologia para melhorar a qualidade dos dados por meio de restrições SHACL geradas a partir das análises feitas pelo ABSTAT, que é uma ferramenta de análise semântica que ajuda os consumidores de dados a entender melhor os dados, extraindo padrões de ontologia orientados por dados e estatísticas. Pandit, O'Sullivan e Lewis (2018) formalizaram a criação de shapes em SHACL por meio da utilização de Padrões de Design de Ontologias (PDO), que são axiomas que capturam apenas os conceitos e relacionamentos necessários para definir determinado domínio. Essa abordagem incentiva a reutilização de PDOs além da fase de modelagem de dados. Os autores propõem que é possível traduzir os axiomas gerados pela PDO em shapes do SHACL, sendo necessária apenas a criação de um mapeamento entre as restrições e os componentes do SHACL.

[1] https://www.w3.org/TR/shacl/#references
[2] https://www.w3.org/RDF/

Rabbani, Lissandrini e Hose (2022) buscaram compreender como os shapes têm sido gerados em SHACL e como têm sido utilizados. Primeiramente, foi realizada uma pesquisa na comunidade, tanto acadêmica quanto empresarial, para analisar as necessidades e comportamentos dos usuários ao gerar shapes SHACL. Os resultados mostraram que métodos automáticos de geração de *shapes* apenas são aplicáveis em pequenos conjuntos de dados. Os resultados também apontaram que nenhum dos shapes gerados por essas ferramentas extrai todas as restrições necessárias.

## 3. Metodologia

Na Figura 1 é possível observar resumidamente os passos metodológicos da pesquisa. Nas próximas seções cada etapa da metodologia é explicada em detalhes e seus resultados são apresentados.

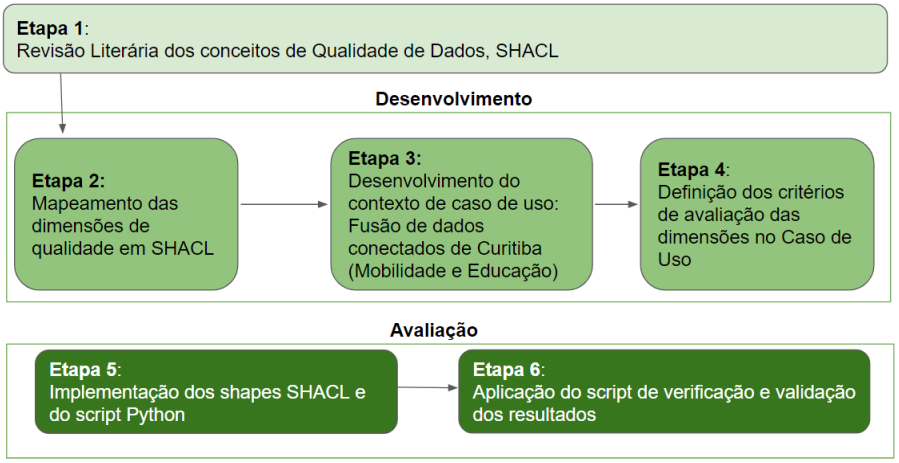


**Figura 1. Metodologia da pesquisa**

## 4. Fundamentação teórica

### 4.1. Dimensões de qualidade de dados

O conceito de qualidade em um conjunto de dados varia de acordo com o contexto em que os dados estão sendo utilizados (WANG; STRONG, 1996). Dados que podem ser úteis para um determinado contexto podem não ter serventia para outro dependendo das características apresentadas por estes dados. Dentro do conceito de qualidade de dados, essas características são chamadas de dimensões (ou critérios). O conjunto de possíveis dimensões que um dado pode ser avaliado é extenso e tanto a escolha de quais dimensões são relevantes quanto como elas devem ser avaliadas estão relacionados ao contexto de uso. Neste trabalho, as dimensões escolhidas para avaliação da qualidade na fusão de dados conectados são *acurácia*, *concisão* e *consistência* devido ao contexto do trabalho estar relacionado à fusão, estas dimensões podem dar conta dos problemas que podem surgir nesta operação com dados conectados.

A *acurácia* tem como objetivo indicar se os dados literais estão de acordo com os padrões definidos pela sua tipagem. Dados literais são representações em string de datas, inteiros e decimais, textos, tempo, etc que podem estar mal formados. Um exemplo de literal malformado é uma data sem caracteres separadores (como

1205/2000) ou um inteiro que contém casas decimais. A dimensão de *concisão* pode ser dividida em dois tipos: de objetos e de atributos. A concisão de objetos, que também pode ser chamada de singularidade, se refere à quantidade de indivíduos repetidos em um determinado conjunto de dados. Por indivíduo repetido se entende duas ou mais representações de um mesmo objeto do domínio em um conjunto de dados (ZAVERI et al., 2013). Por exemplo, dois indivíduos de um conjunto de dados que representa a população do Brasil com o predicado tem CPF 03526411198 (que indica o número do Cadastro de Pessoas Físicas) seriam incoerentes com a realidade do domínio, uma vez que não podem haver duas pessoas com o mesmo número de CPF no Brasil. Assim, a concisão de objetos mede o número de indivíduos únicos em relação ao número total de indivíduos no conjunto de dados. Já a concisão de atributos se refere ao número de predicados únicos de um determinado indivíduo, ou seja, verifica a existência de predicados redundantes nos indivíduos do conjunto de dados [MENDES; MÜHLEISEN; BIZER, 2012]. Por exemplo, um indivíduo que representa um objeto tem dois predicados: dataDeFabricacao e fabricadoEm (no sentido de data e não de local). Os predicados deste indivíduo estão sendo redundantes, uma vez que representam a mesma informação porém com nomes diferentes. Por último, a dimensão de qualidade de *consistência* se refere ao quanto o conjunto de dados está livre de informações conflitantes. Pode-se dizer que um conjunto de dados sem inconsistência é aquele que não contém contradições nos dados com respeito a determinada ontologia (ZAVERI et al., 2013). Um exemplo de dados inconsistentes seria um time de futebol com apenas 5 jogadores (sendo que são necessários no mínimo 11 para uma partida iniciar) ou uma república federativa sem nenhuma federação (que viola o conceito de república federativa).

### 4.2. SHACL

Com o aumento da utilização de dados em RDF, foi necessária a criação de uma linguagem padrão para a criação de restrições de qualidade e de mecanismos que pudessem interpretar essa linguagem e apontar as violações destas restrições. O SHACL (*Shapes Constraint Language*) surgiu com o objetivo de suprir essa demanda e se tornou uma recomendação da W3C em 2017. O SHACL funciona por meio da criação de *shapes*, que são grafos em RDF que definem as restrições para indivíduos específicos. Formalizando, um shape é uma tupla (s, t, d) definida por três componentes: o nome do shape s, que o identifica exclusivamente; os indivíduos a serem validados t, e o conjunto de restrições a serem verificadas d.

## 5. Desenvolvimento

O desenvolvimento refere-se às etapas 2, 3 e 4 da metodologia.

### 5.1. Mapeamento das dimensões em SHACL

Nesta etapa, foram criados os mapeamentos entre os componentes do SHACL e as dimensões de qualidade escolhidas. Os componentes do SHACL são funcionalidades fornecidas pela linguagem, que juntas formam os *shapes* de validação. Como proposto em Pandit, O'Sullivan e Lewis (2018), o mapeamento entre os componentes da linguagem e as dimensões de qualidade deve ser feito para que, posteriormente, as

restrições de qualidade criadas para a ontologia sejam implementadas em SHACL. Para cada dimensão de qualidade, buscou-se entender qual o seu comportamento (como a dimensão é definida) e posteriormente investigar quais funcionalidades do SHACL que suportavam a verificação desta definição no conjunto de dados. A definição das dimensões foi feita com a ajuda de definições formais encontradas na literatura e complementado com observações do autor deste trabalho. O mapeamento pode ser visto na Tabela 1.

**Tabela 1. Mapeamento de dimensões em componentes SHACL**

| Dimensão | Componentes SHACL | Mapeamento |
|---|---|---|
| Acurácia | sh:property / sh:path / sh:dataType/ sh:maxInclusive /sh:minInclusive / sh:pattern / sh:length | Componentes de acesso a predicados e de validação de literais |
| Consistência | sh:property / sh:path /sh:class /sh:minCount /sh:maxCount | Componentes de acesso a predicados e validação de relacionamento entre classes |
| Concisão | sh:closed / sh:ignoredProperties/ sh:property | Componentes de validação de existência de predicados |

## 5.2. O Contexto do Caso de Uso: Dados Conectados de Mobilidade e Educação

O contexto utilizado para os testes são dados abertos conectados de mobilidade e educação da cidade de Curitiba.

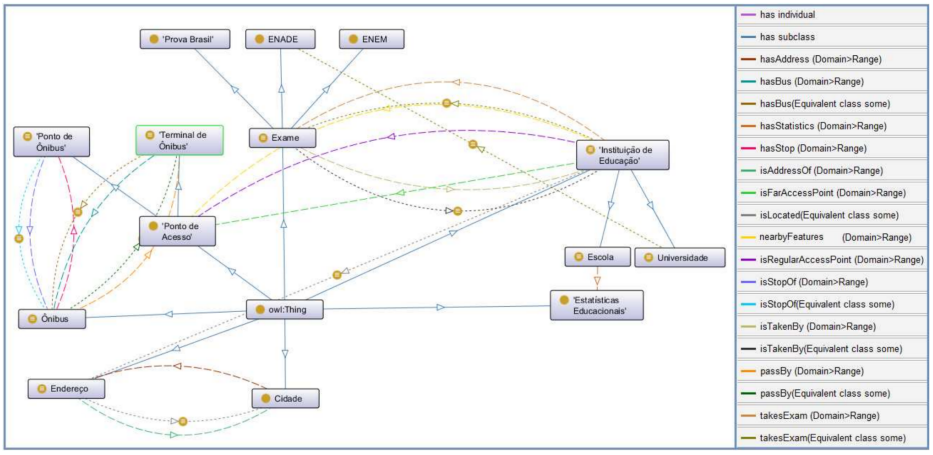


**Figura 2. Ontologia de Mobilidade Urbana e Educação de Curitiba [Belizario, 2020]**

Os dados contêm entidades relacionadas ao transporte público, como pontos de ônibus, linhas de ônibus e terminais de ônibus, bem como as entidades de educação em nível básico, fundamental e superior e por fim as estatísticas de avaliação de cada instituição de ensino, como ENEM, ENADE e Prova Brasil. A ontologia que conecta diferentes

fontes de dados abertos em Curitiba foi criada por Belizario (2022) e pode ser visualizada na Figura 2.

### 5.3. Definição dos critérios de avaliação das dimensões no Caso de Uso

Para a dimensão de acurácia, foram observados quais os valores aceitáveis para determinada variável no domínio. Por exemplo, no caso do ano de realização de um exame educacional não faz sentido esse valor ser menor que 1980 (uma vez que dados muito antigos não devem ser considerados) ou esse valor ser maior que o ano atual (uma vez que é impossível que um exame tenha sido realizado no futuro). Já para o valor do INSE (nível socioeconômico), foi feita uma pesquisa de como esse dado é calculado e quais os possíveis valores que ele pode receber. Por se tratar de um dado do governo, essas informações foram encontradas no site do Ministério da Educação[3].

Para a dimensão de consistência, foi feita uma análise empírica de como as classes se relacionam entre si no domínio da ontologia. Por exemplo, para a criação dos critérios de avaliação da classe *Ônibus*, foi identificado que uma linha de ônibus só faz sentido se parar em pelo menos dois pontos de acesso (ponto de ônibus ou terminal). Um ônibus que pára em apenas um lugar não faz sentido, pois os passageiros irão desembarcar no mesmo lugar que embarcaram, assim como um ônibus sem paradas não teria como embarcar passageiros. Ou seja, a partir da análise dos axiomas das classes, foi possível determinar como elas se comportam no domínio e consequentemente quais incoerências ferem a essência desse objeto.

Para a dimensão de concisão de objetos, foram analisadas quais características dos dados seriam únicas a cada indivíduo, ou seja, que não poderiam se repetir no domínio. Pelo contexto da ontologia do trabalho, foi utilizado um código de identificação para pontos de acesso, linhas de ônibus e instituições educacionais. Logo, foi escolhido o predicado *hasCode* para a verificação de duplicatas nos dados. Já para a concisão de atributos, foram considerados os predicados que já estavam definidos na ontologia. Ou seja, se um predicado que não está na ontologia for utilizado, haverá a quebra da restrição. Na Tabela 2 estão listados os critérios de avaliação das classes.

**Tabela 2. Definição dos critérios para avaliação das dimensões na ontologia**

| **Entidade na ontologia Mobilidade e Educação** | **Critério de avaliação das dimensões** |
|---|---|
| classe “Ônibus” | Deve ter um nome que seja uma string de letras (*acurácia*); Deve ter um código que deve ser único (*concisão*); Deve parar em pelo menos dois pontos OU deve passar em pelo menos dois terminais<br>OU deve passar em pelo menos um ponto e um terminal (*consistência*) |

[3] https://www.gov.br/inep/pt-br/acesso-a-informacao/dados-abertos/indicadores-educacionais/nivel-socioeconomico

| | |
|---|---|
| classe “Ponto de Acesso” | Deve ter um código que deve ser único (*concisão*); Deve estar localizada em algum endereço (*consistência*) |
| classe “Ponto de ônibus” | Herda os critérios da classe “Ponto de Acesso”, que é a classe pai de “Ponto de ônibus” e contém mais o seguinte critério: Deve ser parada de pelo menos um ônibus (*consistência*) |
| classe “Terminal de ônibus” | Herda os critérios da classe “Ponto de Acesso”, que é a classe pai de “Terminal de ônibus” e contém mais o seguinte critério: Deve ter pelo menos um ônibus passando (*consistênci*a) |
| classe "Instituição de Educação” | • Deve ter um nome que seja uma string de letras (*acurácia*); Deve ter um tipo, que é uma string que pode ter os valores “Particular”, “Público Municipal”, “Público Estadual”, “Público Federal” (*acurácia*); Deve ter código INEP (código que identifica unicamente uma instituição), que deve ter 8 números (*acurácia*); Pode conter pontos de acessos (*consistência*); Deve fazer pelo menos um exame (*consistência*); Deve estar localizada em algum endereço (*consistência*) |
| classe “Universidade” | Herda os critérios da classe “Instituição de Educação”, que é a classe pai de “Universidade” e contém mais os seguintes critérios: Deve ter código IES (código que identifica a Instituição de Ensino Superior), que deve ter de 3 a 5 dígitos *hasCodeIES*; Deve ter iniciais que é uma string de letras e símbolos *hasInitials* |
| classe “Escola” | Herda os critérios da classe "Instituição Educação”, que é a classe pai de “Escola e contém mais o seguinte critério: Deve ter pelo menos uma estatística vinculada (*consistência*) |
| classe “Estatísticas Educacionais” | Pode conter taxa de abandono, que deve ser um número float (*acurácia*); Pode conter taxa de aprovação, que deve ser um número float (*acurácia*); Pode conter taxa de reprovação, que deve ser um número float (*acurácia*); Pode conter taxa de permanência, que deve ser um número float (*acurácia*); Deve ter um ano, que é um inteiro entre 1980 e 2022 (acurácia); Deve ter um INSE, que deve ser um inteiro entre 1 e 9 (*acurácia*) |
| classe “Exame” | Deve ter um INSE, que deve ser um inteiro entre 1 e 9 (*acurácia*) |

| | |
|---|---|
| classe “Endereço” | Deve ter um nome que seja uma string de letras (*acurácia*);<br>Deve ter um bairro que seja uma string de letras (*acurácia*) |

### 5.3. Implementação dos critérios em SHACL e Aplicação do Script

As etapas 5 e 6 da metodologia dizem respeito à avaliação do que foi desenvolvido. Uma vez feito o mapeamento dos componentes do SHACL e com a criação dos critérios para cada classe da ontologia, foi possível fazer a tradução dos critérios de linguagem natural (Tabela 2) para a linguagem SHACL. A Figura 3 é um exemplo da implementação dos critérios para classe “Ponto de Acesso” seguindo o mapeamento dos componentes realizado na Tabela 1 e as restrições em linguagem natural na Tabela 2.

```
:AccessPointShape a sh:NodeShape ;
    sh:targetClass :Access_Point ;
    sh:property [
        sh:message '''{"message": "Access Point has invalid code.", "type": "property"}''' ;
        sh:path :hasCode ;
        sh:minCount 1 ;
        sh:maxCount 1 ;
        sh:minInclusive 1 ;
        sh:maxInclusive 9999 ;
        sh:datatype xsd:integer ;
    ] ;
    sh:property [
        sh:message '''{"message": "Access Point has invalid address.", "type": "property"}''' ;
        sh:path :isLocated ;
        sh:minCount 1 ;
        sh:maxCount 1 ;
        sh:class :Address ;
    ] ;
    sh:closed true ;
    sh:ignoredProperties (rdf:type) ;
.
```

**Figura 3. Shape para avaliação da Classe “Ponto de Acesso”**

No total, foram criados dez shapes em SHACL, sendo nove específicos para cada classe da ontologia e mais um que é reutilizado nas classes “Instituição de Educação”, “Ônibus” e “Ponto de Ônibus”. Além dos shapes em SHACL, foi necessária a criação de um script em Python para permitir a execução do SHACL em cima de uma ontologia e de um conjunto de dados. Além disso, o script foi estendido para permitir a validação da dimensão de qualidade de concisão. Neste script são utilizadas as bibliotecas pyshacl (para a validação das restrições), rdflib (para a manipulação de dados em RDF), xml (para a manipulação de dados em XML) e json. Os shapes e o script em Python podem ser vistos em https://github.com/otaviobertucini/SHACL-validator/tree/master.

## 6. Resultados

Para a avaliação da verificação das 3 dimensões de qualidade implementadas em SHACL foram utilizados dois conjuntos de dados, um contendo dados sem erro (conjunto A) contém pelo menos uma instância de cada classe da ontologia e é referente ao transporte público de Curitiba e educação, que seriam os dados originais; e outro (conjunto B) contendo dados adicionais com algumas irregularidades inseridas de propósito, como um ônibus que não passa em nenhum ponto, um ponto de ônibus que não tem nenhuma linha, uma universidade que contém alguns dados mal formatados e indivíduos duplicados.

Primeiramente, o script de validação em Python foi executado utilizando o conjunto de dados A. Uma vez verificado que o conjunto de dados não contém nenhuma quebra de restrições que foram definidas em SHACL, o conjunto A e o conjunto B foram fundidos utilizando o VSCode e posteriormente o script de validação em Python foi executado utilizando o conjunto resultante da fusão. O objetivo desta etapa é verificar se as quebras de restrições que foram inseridas no conjunto de dados através da

fusão foram identificadas pelo script. Feito isso, o script foi executado novamente, porém agora com os indivíduos duplicados ligados através do predicado olw:sameAs, que indica que dois indivíduos representam o mesmo objeto do domínio. Após a execução, foi validado se os casos de duplicação identificados pelas restrições de concisão de objetos foram ignorados nas situações onde os objetos continham o predicado *owl:sameAs* entre eles.

Através dos testes realizados, mostrou-se que o script desenvolvido para a verificação das dimensões de qualidade de acurácia, concisão e consistência valida corretamente os dados da ontologia através dos shapes do SHACL e ignora os casos de duplicação de predicados únicos quando os indivíduos duplicados estão ligados pelo predicado *owl:sameAs*, uma vez que foram mostradas para o usuário todas as restrições definidas no SHACL que foram quebradas, com exceção daquelas que não se tratavam de duplicatas. Após a execução do script de validação, a lista de quebras de restrições foi apresentada no terminal. Cada quebra de restrição possui duas linhas, onde a primeira linha do erro indica qual o tipo do erro e a segunda em qual indivíduo ocorreu a quebra da restrição. Além disso, após adicionar o predicado *owl:sameA*s entre os indivíduos duplicados, as quebras de restrição de concisão de objetos referentes a esses indivíduos foram ignoradas e não apareceram no terminal após a execução. Isso comprova que o script desconsidera corretamente os indivíduos duplicados mas que representam a mesma entidade no mundo real. As Figuras 4 e 5 mostram as restrições de qualidade sendo verificadas e detectadas.

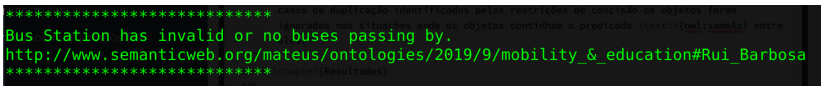


**Figura 4. Erro de consistência na Classe “Ponto de Ônibus”**

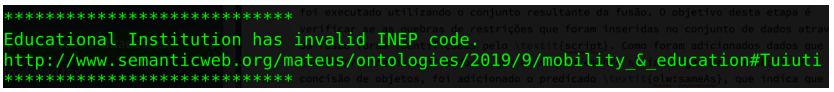


**Figura 5. Erro de acurácia na classe “Instituição de Educação”**

## 7. Conclusão

Neste trabalho foram apresentados um dos potenciais problemas gerados pela fusão de dois ou mais conjuntos de dados conectados, que é a quebra das dimensões de qualidade de consistência, concisão e acurácia. Esse trabalho propôs a validação dessas dimensões após a fusão por meio do uso de SHACL. Um estudo de caso foi realizado com dados conectados dos domínios de mobilidade urbana e educação da cidade de Curitiba. Os resultados mostram que, por meio da ontologia de validação em SHACL, foi possível identificar as quebras nas restrições de qualidade no conjunto de dados fundido. Além disso, esse trabalho formaliza a validação das três dimensões escolhidas através da linguagem SHACL, permitindo com que outros trabalhos utilizem o mapeamento feito para criarem seus próprios *shapes* de validação em SHACL. As cidades inteligentes se beneficiam de propostas como estas desde que invistam em formatos de grafos de conhecimento em suas plataformas de dados abertos. Como proposta de trabalhos futuros, este trabalho pode ser estendido para que usuários que desejem criar *shapes* no

SHACL para as dimensões tratadas neste trabalho não precisem utilizar a linguagem SHACL diretamente. No lugar, pode ser criada uma interface gráfica onde o usuário pode criar as restrições que posteriormente seriam mapeadas através dos mapeamentos feitos neste trabalho.

## Referências


Belizario, M, G., Rita Cristina G. Berardi. Linked Open Data in Smart Cities: An application in the domains of Mobility and Education. Anais do XVII Escola Regional de Banco de Dados. SBC, 2022.

Halevy, A., Norvig, P., & Pereira, F. (2009). The unreasonable effectiveness of data. IEEE intelligent systems, 24(2), 8-12..

Heath, T.; Bizer, C. (2011) "Linked data: Evolving the web into a global data space. Synthesis lectures on the semantic web: theory and technology", Morgan & Claypool Publishers, v. 1, n. 1, p. 1–136, 2011.

Mendes, P. N., Mühleisen, H., & Bizer, C. (2012, March). Sieve: linked data quality assessment and fusion. In Proceedings of the 2012 joint EDBT/ICDT workshops (pp. 116-123).

Naphade, M., Banavar, G., Harrison, C., Paraszczak, J., & Morris, R. (2011). Smarter cities and their innovation challenges. Computer, 44(6), 32-39.

Pandit, H. J., O'Sullivan, D., & Lewis, D. (2018, October). Using Ontology Design Patterns To Define SHACL Shapes. In WOP@ ISWC (pp. 67-71).

Paulheim, H., & Stuckenschmidt, H. (2016). Fast approximate a-box consistency checking using machine learning. In The Semantic Web. Latest Advances and New Domains: 13th International Conference, ESWC 2016, Heraklion, Crete, Greece, May 29--June 2, 2016, Proceedings 13 (pp. 135-150). Springer International Publishing.

Rabbani, K., Lissandrini, M., & Hose, K. (2022, April). SHACL and ShEx in the Wild: A Community Survey on Validating Shapes Generation and Adoption. In Companion Proceedings of the Web Conference 2022 (pp. 260-263).

Rietveld, L., Beek, W., & Schlobach, S. (2015). LOD lab: Experiments at LOD scale. In The Semantic Web-ISWC 2015: 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part II 14 (pp. 339-355). Springer International Publishing.

Spahiu, B., Maurino, A., & Palmonari, M. (2018, October). Towards Improving the Quality of Knowledge Graphs with Data-driven Ontology Patterns and SHACL. In ISWC (Best Workshop Papers) (pp. 103-117).

Zaveri, A., Kontokostas, D., Sherif, M. A., Bühmann, L., Morsey, M., Auer, S., & Lehmann, J. (2013, September). User-driven quality evaluation of dbpedia. In Proceedings of the 9th International Conference on Semantic Systems (pp. 97-104).

Wang, R. Y., & Strong, D. M. (1996). Beyond accuracy: What data quality means to data consumers. Journal of management information systems, 12(4), 5-33.