

# LiRB: Um Navegador Leve Baseado em Texto para Knowledge Graphs RDF

Caio Viktor S. Avila<sup>1</sup>, Vânia Vidal<sup>1</sup>

<sup>1</sup>Departamento de Computação – Universidade Federal do Ceará (UFC)  
Campus do Pici – Bloco 910 – Fortaleza – CE – Brazil

caioviktor@aluf.ufc.br, vvidal@lia.ufc.br

**Abstract.** *In this article we present Light Rdf Browser (LiRB), a lightweight interface for interactive browsing over Knowledge Graphs (KGs) RDF. LiRB features a text-based Web interface, providing a navigation experience similar to traditional Web pages, promoting a lower learning curve for users. As differences, LiRB uses simple and fast queries, being ideal for the browsing of virtual KGs. In addition, the tool also allows the abstraction of N-ary relationships to represent the semantics of relationship's properties. Finally, LiRB also supports the feature of displaying events Timelines of resource in the KG.*

**Resumo.** *Neste artigo apresentamos Light RDF Browser (LiRB<sup>1</sup>), uma interface leve para a navegação interativa sobre Knowledge Graphs (KGs) RDF. LiRB apresenta uma interface Web baseada em texto, proporcionando uma experiência de navegação semelhante às páginas Web tradicionais, promovendo uma menor curva de aprendizagem aos usuários. Como diferenciais, LiRB utiliza consultas simples e rápidas, sendo ideal para a navegação de KGs massivos e/ou virtuais. Além disso, a ferramenta também permite a abstração de relacionamentos N-ários para representar a semântica de propriedades de relacionamentos, além da apresentação de consultas salvas. Por fim, LiRB também dá suporte ao recurso de exibição de timelines de eventos de recursos do KG.*

## 1. Introdução

*Knowledge Graphs* (KGs) são conjuntos de dados que seguem um modelo baseado em grafos para capturar conhecimento em cenários que envolvam integração, gerenciamento e extração de valor de diversas fontes de dados em grande escala [Noy et al. 2019]. *RDF Browser* ou *Semantic Web Browser* são aplicações que localizam automaticamente metadados e montam interfaces de apontar e clicar a partir de especificações ontológicas descritas em RDF e recuperadas dinamicamente da Web Semântica [Quan and Karger 2004]. Deste modo, tais ferramentas podem ser reutilizadas no contexto de KGs armazenados no formato RDF.

Embora interfaces baseadas em técnicas de visualização de grafos, tais como as apresentadas em [Bikakis and Sellis 2016], tenham um forte apelo, o uso destas na prática pode vir a ser um desafio quando navegando sobre grafos com um grande número de nós, gerando uma visualização poluída e de difícil compreensão. Além disso, interfaces deste tipo também tendem a apresentarem consultas mais complexas, o que pode vir a impactar

---

<sup>1</sup>Vídeo de demonstração: <https://youtu.be/MQrRGVzfEGI>

negativamente no tempo de renderização de KGs massivos e/ou virtuais. KGs virtuais reescrevem consultas SPARQL para consultas SQL sobre a base de dados relacional original dos dados, onde em geral, relações entre dois recursos no KG são traduzidas para operações de junções SQL [Rodriguez-Muro et al. 2013]. Deste modo gerando um elevado número de junções para consultas SPARQL mais complexas, o que ocasiona um elevado custo para o processamento da consulta SQL resultante.

Deste modo, surge a necessidade do desenvolvimento de *RDF Browsers* modernos que apresentem os dados de maneira compreensível e que utilizem consultas leves, compatíveis com a possibilidade de KGs virtuais. Portanto, neste artigo apresentamos *Light RDF Browser* (LiRB), uma interface leve para a navegação interativa sobre KGs RDF. LiRB apresenta uma interface Web baseada em texto, proporcionando uma experiência de navegação semelhante às páginas Web tradicionais, promovendo uma menor curva de aprendizado aos usuários. Como diferenciais, LiRB utiliza consulta simples e rápidas, sendo ideal para a navegação de KGs virtuais. Além disso, a ferramenta também permite a abstração de relacionamentos *N-ários* para representar a semântica de propriedades de relacionamentos. Ademais, a ferramenta também permite que o desenvolvedor salve consultas pré-definidas a serem apresentadas ao usuário. Por fim, LiRB também dá suporte ao recurso de exibição de *timelines* de eventos de recursos do KG.

O restante deste artigo é estruturado com a seguir. Na seção 2 são apresentados o trabalhos relacionados. Na seção 3 é apresentada a interface de usuário da ferramenta LiRB. Por fim, na seção 4 são apresentadas as conclusões dos autores.

## 2. Trabalhos Relacionados

Em [Rutledge et al. 2005] os autores apresentam *Noadster*, um sistema independente de domínio para a geração automática de páginas HTML de dados RDF. *Noadster* foi um dos primeiros a tentar se aproximar da visão de um *RDF browser* universal, permitindo a criação de visualizações independentes de domínio, em contraponto às visualizações construídas manualmente para cada recurso RDF costumeiramente adotadas na época. Embora *Noadster* tenha apresentado-se como uma ideia promissora, a abordagem de *RDF Browsers* baseados em texto foi abandonada em detrimento de abordagens baseadas em técnicas de visualização de grafos [Beck et al. 2017].

[Camarda et al. 2012] apresentam *LodLive*, uma das mais populares aplicações para a navegação de recursos RDF. *LodLive* apresenta quinze diferentes endpoints SPARQL prontos para usar, permitindo ao usuário adicionar o seu próprio, além de customizar a interface. *LodLive* apresenta um estilo de navegação visual, representando recursos RDF como nós de um grafo e relacionamentos (*owl:ObjectProperties*) como arestas entre estes nós, permitindo a expansão dos nós de maneira iterativa. No entanto, assim como outros *RDF Browser* visuais, a navegação em grafos massivos pode tornar-se poluída e lenta, por conta do grande número de nós e arestas.

Em [Haase et al. 2019] é apresentada a plataforma *metaphactory*, uma plataforma para criar aplicativos de gerenciamento de KGs. Dentre uma de suas muitas funções, a plataforma permite que o desenvolvedor disponibilize uma interface de navegação visual. A plataforma permite que o desenvolvedor personalize a interface de navegação, além de permitir sua integração com outras ferramentas da mesma, tais como ferramentas de busca, edição e gerenciamento de dados, etc. No entanto, embora ela permita uma grande

customização da interface, seu uso depende da adoção de uma grande estrutura, o que pode ser uma desvantagem em cenários que exijam soluções mais simples. Embora haja uma versão *stand-alone* da aplicação de navegação, chamada de Ontodia, esta possui as mesmas limitações de demais interfaces do tipo, ao lidar com grafos massivos ou virtuais.

Deste modo, LiRB surge como uma opção leve, independente de domínio e com funcionalidades avançadas para a navegação de KGs RDF. LiRB é uma solução leve tanto para o servidor hospedando a aplicação, não requerendo a instalação de grandes pacotes de software, quanto para o *endpoint SPARQL* hospedando o KG, realizando consulta simples e leves, ideais para grafos massivos e/ou virtuais. Além disso, por tratar-se de uma interface baseada em texto, LiRB evita o problema de apresentar um grande número de nós e arestas de maneira desorganizada, apresentando os valores de propriedades como listas recolhíveis de itens.

### 3. LiRB: Light RDF Browser

O código fonte da aplicação é disponibilizado de maneira aberta através do repositório presente no link<sup>2</sup>. A ferramenta foi desenvolvida com o uso da linguagem python, utilizando o *framework Flask* para prover um serviço web leve, deste modo podendo ser hospedado em qualquer *host* dotado com um interpretador python. Além disso, a ferramenta pode ser acessada a partir de qualquer dispositivo com acesso à rede através de um *web browser*.

LiRB atua como uma interface sobre *endpoints SPARQL*, realizando apenas consultas em SPARQL padrão para acessar os dados, desta maneira podendo ser facilmente adotado em qualquer projeto como um *RDF browser* para qualquer KG RDF. No link<sup>3</sup> encontra-se disponível um KG RDF simples que será utilizado como demonstração da ferramenta. Para acessar a demonstração, basta carregar o KG de demo um qualquer *triplestore* e configurar o arquivo de configurações da ferramenta como indicado no repositório do projeto. Neste artigo foi considerado o *triplestore* GraphDB. A seguir são apresentados os principais modos de visualização de KGs RDF por LiRB.

#### 3.1. Navegação de Recursos

A tela inicial apresentada ao acessar a ferramenta é a tela de seleção de classes (Figura 1). Nesta tela são apresentadas as classes existentes no grafo acompanhadas de seus rótulos (P1), *URIs* (P2), descrições (P3) e sua lista de sub-classes (P4). Nesta tela as classes são dispostas em duas listas: (L1) a lista de classes em destaques, onde são apresentadas as classes marcadas com o atributo “*lirb:has\_spotlight “true”*” ou através da definição da lista no arquivo de configurações. (L2) a lista geral de classes, onde são apresentadas todas as classes definidas no KG.

Após selecionar uma classe (Ao clicar em seu respectivo botão “Explore”), é apresentada a lista de recursos pertencentes à classe selecionada. Em qualquer uma das duas telas é possível pesquisar recursos pela *rdfs:label* ou por trechos da URI, no entanto, quando a busca é realizada na tela de lista de recursos de uma classe específica, a busca é limitada para recursos da classe selecionada.

<sup>2</sup><https://github.com/CaioViktor/LiRB>

<sup>3</sup>[https://github.com/CaioViktor/LiRB/blob/main/ontology\\_example.ttl](https://github.com/CaioViktor/LiRB/blob/main/ontology_example.ttl)

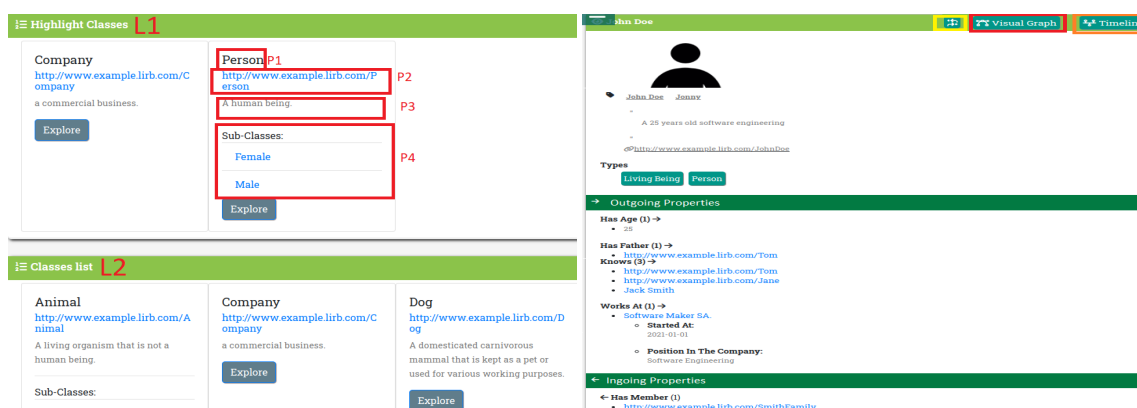


Figura 1. Tela de seleção de classes

Figura 2. Tela de navegação de recurso

Ao selecionar um recurso da tela de lista de recursos, LiRB apresenta a tela de navegação de recursos (Figura 2). Nesta tela são apresentadas as propriedades com as quais o recurso está envolvido no KG, seja como sujeito de triplas (*outgoing properties*, ou propriedades partido do recurso atual) ou como objeto de uma tripla (*ingoing properties*, ou propriedades chegando no recurso atual). No caso de propriedades *owl:ObjectProperty* (relacionamentos entre dois recursos), a interface permite a navegação iterativa entre os recursos (apresentados como *links html*) envolvidos nestas. A Figura 2 apresenta as propriedades do recurso “John Doe”, que por sua vez possui a propriedade “Has Father”, relacionando-o ao recurso “Tom”. Ao clicar na URI (ou *rdfs:label* quando disponível) a interface será redirecionada para a página exibindo as propriedades do recurso “Tom”.

Além disso, LiRB oferece a possibilidade de uma fácil integração com *RDF Browsers* visuais, permitindo o redirecionamento para a navegação visual ao clicar no botão destacado em vermelho (Figura 2). Por fim, LiRB também permite que o usuário ative (botão destacado em amarelo na Figura 2) a visualização de propriedades de outras versões do recurso selecionado indicadas por links *owl:sameAs*.

Além de relações simples representadas por uma única tripla RDF, LiRB também dá o suporte para relações complexas comumente chamadas de *N-ary Relations*. Relacionamentos N-ary (N-ários ou enários) são relacionamentos onde um sujeito está relacionado a mais de um objeto por meio de uma única propriedade ou quando o próprio relacionamento entre dois recursos possui atributos [Giunti et al. 2021]. Na abordagem seguida<sup>4</sup> a propriedade que representa um relacionamento N-ário relaciona-se com um recurso intermediário sem valor semântico, mas que serve para agregar os n valores complementares do relacionamento. LiRB adota o uso de uma marcação por meio da classe “*lirb:N\_ary\_Relation\_Class*” para abstrair este tipo de relacionamento, apresentando sua semântica correta. A Figura 2 apresenta um relacionamento N-ário na propriedade “*Works At*”, onde o valor da propriedade é apresentado como o recurso “*Software Maker S.A.*”, acompanhado dos metadados da relação (atributos da relação) “*Started At*” e “*Position in The Company*”.

<sup>4</sup><https://www.w3.org/TR/swbp-n-aryRelations/>

### 3.2. Timelines de Recursos

Além da navegação tradicional, LiRB também oferece a possibilidade da visualização de *Timelines* de recursos. A tela de *timeline* de um recurso é acessível quando este possuir a propriedade “*tlo:has\_timeLine*” (acessível através do botão “Timeline” destacado em laranja na Figura 2). A tela de exibição de *timelines* (Figura 3) apresenta todos os eventos sofridos pelo recurso selecionado dispostos de maneira agrupada por data e em ordem cronológica. Deste modo, a *timeline* é separada em instantes, onde cada instante possui uma data (no formato dia/mês/ano hora:minutos:segundos) e sua lista de atualizações. Para atualizações de atributos (*owl:DatatypeProperty*) são apresentados os valores antigo e novo atualizados na data (marcação número 2 na Figura 3). Além disso, LiRB também permite filtrar a *timeline* por uma propriedade específica, apresentando apenas eventos daquele tipo de propriedade (marcação em laranja na Figura 3).



ID	Tipo de evento	Classe do evento
1	Inserção do recurso	tlo:Insertion
2	Atualização de atributo ( <i>owl:DatatypeProperty</i> )	tlo:Update_Property
3	Inserção de relacionamento ( <i>owl:ObjectProperty</i> )	tlo:Insert_Relationship
4	Remoção de relacionamento	tlo:Remove_Relationship
5	Remoção de recurso	tlo:Remotion

Tabela 1. Tipos de eventos de *time-line*

Figura 3. Visualização de *Timeline*

A anotação da *timeline* de um recurso é realizada seguindo uma ontologia que não será apresentada neste trabalho por conta da limitação de espaço. Deste modo, LiRB dá suporte a 5 diferentes tipos de eventos de *timeline* como apresentado na Tabela 1. O campo *ID* na Tabela 1 representa o identificador do tipo de evento que é utilizado para marcar (numeração vermelha) os exemplos de eventos deste tipo na Figura 3. Tendo como objetivo garantir um bom desempenho da ferramenta, as anotações de *timelines* podem ser armazenadas em um repositório diferente do da A-Box.

### 3.3. Consultas Salvas

Além dos modos de visualização anteriormente citados, LiRB também conta com a possibilidade de apresentar consultas previamente salvas. Esta funcionalidade permite que o desenvolvedor salve consultas específicas, para que os resultados destas possa ser visto através da interface pelos usuários.

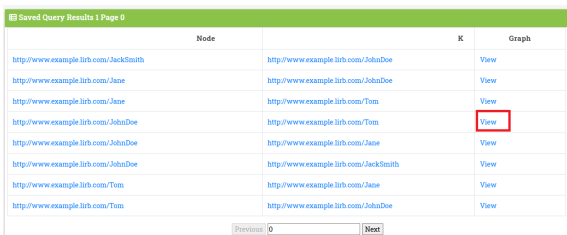


Figura 4. Resultado de consulta salva

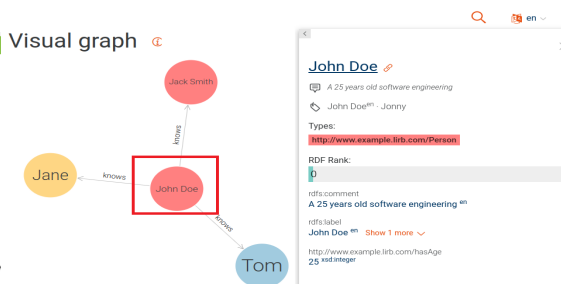


Figura 5. Visualização do recorte do grafo de consulta salva

A Figura 4 apresenta o resultado de uma consulta salva que recupera todas as triplas sobre uma pessoa conhecer outra. A Figura 5 apresenta um grafo composto por todos os conhecidos de “John Doe” ao clicar no link destacado em vermelho na Figura 4. O recorte a ser apresentado pode ser definido pelo desenvolvedor por meio de consultas CONSTRUCT.

#### 4. Conclusão

Neste artigo foi apresentada a ferramenta de navegação de KGs RDF baseada em texto *Light RDF Browser* (LiRB). LiRB é uma ferramenta que tenta unir a leveza e agilidade dos primeiros *RDF Browsers* baseados em textos, ponto especialmente relevante em grafos massivos e/ou virtuais, com funcionalidades modernas.

Para tal, LiRB disponibiliza as opções de navegação de recursos com abstrações de relações *N-árias*, visualização de *timelines* e consultas salvas acompanhadas da navegação visual de recortes do KG que as represente. Para todas estas funcionalidades LiRB utiliza apenas consultas SPARQL leves que ainda podem ser simplificadas via configurações, desta maneira permitindo que o desenvolvedor escolha o balanço entre funcionalidades e responsividade da interface que mais se adéque à realidade de seu KG.

#### References

- Beck, F., Burch, M., Diehl, S., and Weiskopf, D. (2017). A taxonomy and survey of dynamic graph visualization. In *Computer graphics forum*, volume 36, pages 133–159. Wiley Online Library.
- Bikakis, N. and Sellis, T. (2016). Exploration and visualization in the web of big linked data: A survey of the state of the art. *arXiv preprint arXiv:1601.08059*.
- Camarda, D. V., Mazzini, S., and Antonuccio, A. (2012). Lodlive, exploring the web of data. In *Proceedings of the 8th International Conference on Semantic Systems*, pages 197–200.
- Giunti, M., Sergioli, G., Vivanet, G., and Pinna, S. (2021). Representing n-ary relations in the semantic web. *Logic Journal of the IGPL*, 29(4):697–717.
- Haase, P., Herzig, D. M., Kozlov, A., Nikolov, A., and Trame, J. (2019). metaphactory: A platform for knowledge graph management. *Semantic Web*, 10(6):1109–1125.
- Noy, N., Gao, Y., Jain, A., Narayanan, A., Patterson, A., and Taylor, J. (2019). Industry-scale knowledge graphs: Lessons and challenges: Five diverse technology companies show how it’s done. *Queue*, 17(2):48–75.
- Quan, D. and Karger, R. (2004). How to make a semantic web browser. In *Proceedings of the 13th international conference on World Wide Web*, pages 255–265.
- Rodriguez-Muro, M., Kontchakov, R., and Zakharyashev, M. (2013). Ontology-based data access: Ontop of databases. In *The Semantic Web–ISWC 2013: 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21–25, 2013, Proceedings, Part I 12*, pages 558–573. Springer.
- Rutledge, L., van Ossenbruggen, J., and Hardman, L. (2005). Making rdf presentable: integrated global and local semantic web browsing. In *Proceedings of the 14th international conference on World Wide Web*, pages 199–206.