

Representação e Processamento de Conhecimento na Web

Construção de uma Ontologia sobre jogos digitais

João Ribeiro
(PG57565)

Benjamim Rodrigues
(PG57511)

Pedro Pereira
(PG57599)

31 de maio de 2025

1 Introdução

Este projeto teve como objetivo criar uma ontologia completa sobre jogos digitais, suportada por uma aplicação web desenvolvida com Flask. O nosso foco principal foi representar de forma semântica os dados extraídos automaticamente da *DBpedia*, o que permite uma navegação estruturada e intuitiva através de entidades como jogos, plataformas, géneros, desenvolvedoras e publicadores.

Para recolher os dados, utilizámos queries *SPARQL*, seguidas de filtragem e tratamento das inconsistências encontradas. Depois, geramos uma ontologia *OWL* serializada em formato Turtle (TTL) com a biblioteca *rdflib* do Python.

Com este trabalho, pretendemos não apenas organizar a informação de forma mais compreensível e navegável, mas também demonstrar como a Web Semântica pode ser aplicada de forma prática na área dos jogos digitais. A aplicação inclui funcionalidades interativas como pesquisa, páginas de detalhe para cada entidade e um modo de quiz para testar conhecimentos de forma divertida.

2 Arquitetura Geral

A arquitetura do nosso sistema é composta por três componentes principais: extração de dados, modelação ontológica e interface web.

O primeiro módulo, `dbpedia.py`, executa queries *SPARQL* no endpoint público da *DBpedia*. Estas queries recolhem informações essenciais sobre jogos digitais (título, data de lançamento, géneros, plataformas, desenvolvedores e publicadores).

Os dados recolhidos são guardados num ficheiro JSON intermediário, que serve de base para o script `povoar.py`. Este segundo módulo é responsável por converter o dataset numa ontologia completa em *OWL*, estruturada com classes e propriedades bem definidas, seguindo os princípios da modelação semântica.

Finalmente, a aplicação web `app.py` oferece uma interface acessível ao utilizador, suportada por templates Jinja2. Esta interface permite explorar a ontologia através de visualizações detalhadas, pesquisa textual e um jogo de perguntas interativo. Todo o sistema foi desenvolvido com modularidade em mente, facilitando futuras alterações e expansões.

3 Modelo de Ontologia

A ontologia que definimos organiza a informação em torno de cinco entidades principais:

- **Jogo**
- **Plataforma**
- **Género**
- **Desenvolvedora**
- **Publicador**

Cada uma destas entidades possui propriedades específicas que descrevem as suas características e relações com outras entidades. Por exemplo, a classe Jogo tem propriedades como título, data de lançamento e descrição textual. Além disso, relaciona-se com outras entidades através de propriedades como:

- `foiDesenvolvidoPor`
- `foiPublicadoPor`
- `pertenceAoGenero`
- `disponivelEm`

Para permitir navegação bidirecional, todas estas relações são complementadas por propriedades inversas:

- `desenvolveu`
- `publicou`
- `temJogo`
- `suporta`

O modelo segue os princípios da linguagem *OWL*, incluindo a definição explícita de domínios e alcances (domain e range) e a separação clara entre propriedades literais e objetuais. Os relacionamentos muitos-para-muitos (como entre jogos e plataformas) foram cuidadosamente modelados com múltiplas instâncias e *URIs* únicas para cada entidade.

4 Extração e Processamento de Dados

A recolha de dados foi realizada a partir da *DBpedia*, utilizando múltiplas queries *SPARQL* para obter jogos com título e data de lançamento válidos. Usámos o endpoint público <https://dbpedia.org/sparql>, e implementámos um sistema de paginação com blocos de 1000 resultados e um limite máximo de 20 páginas, usando os parâmetros `LIMIT` e `OFFSET`. Fizemos isto uma vez que a *DBpedia* tem um limite imposto de 10000 buscas por query.

Para cada jogo identificado, executámos uma query detalhada que extraiu os seguintes campos:

- Título do jogo (`foaf:name`)
- Data de lançamento (`dbo:releaseDate`)
- Descrição (`dbo:abstract`)
- Géneros (`dbo:genre`)
- Plataformas (`dbo:computingPlatform` ou `dbp:platforms`)
- Desenvolvedores (`dbo:developer` ou `dbp:developer`)
- Publicadores (`dbo:publisher` ou `dbp:publisher`)

Para lidar com múltiplos valores por campo (como géneros ou plataformas), utilizámos o operador `GROUP_CONCAT` com o separador `"`. Isto permitiu converter facilmente os resultados em listas de Python durante o processamento.

O processamento dos dados incluiu várias etapas de limpeza:

- Eliminação de entradas incompletas ou sem *URI* válida
- Conversão de datas para o formato ISO (YYYY-MM-DD)
- Separação das strings concatenadas em listas estruturadas

O grafo final foi serializado em formato Turtle, resultando no ficheiro `games.ttl`, que serve de base para toda a aplicação web. O dataset final contém mais de 15.000 jogos com informações suficientemente completas para alimentar a interface, sistema de pesquisa e modo quiz.

5 Funcionalidades da Aplicação Web

A aplicação web foi desenvolvida com o uso de microframework Flask, seguindo uma arquitetura modular baseada no padrão MVC. O sistema proporciona uma interface intuitiva que permite aos utilizadores navegar pela ontologia de forma fluida e interativa.

As principais funcionalidades incluem:

- Listagem completa de jogos e entidades relacionadas (plataformas, géneros, desenvolvedoras, publicadores)
- Sistema de pesquisa textual por título de jogo, com suporte a correspondência parcial e insensível a maiúsculas/minúsculas
- Visualização detalhada de cada entidade, apresentando atributos e relações associadas de forma clara
- Modo de quiz interativo, com perguntas geradas dinamicamente a partir da ontologia

Cada entidade possui uma página dedicada onde são apresentados os seus dados principais (como nome e data) e ligações clicáveis para outras entidades relacionadas. Isto permite uma exploração progressiva e natural da informação.

O modo de quiz destaca-se como uma funcionalidade educativa e divertida. As perguntas são geradas automaticamente com base no grafo *RDF*, incluindo questões como "Qual destas plataformas está associada ao jogo X?" ou "Quem desenvolveu o jogo Y?". As respostas são avaliadas em tempo real, mantendo um contador de perguntas certas e erradas durante cada sessão.

A aplicação está preparada para lidar com situações de erro e entidades inexistentes, apresentando mensagens claras e informativas ao utilizador. O sistema foi concebido com foco na escalabilidade, permitindo facilmente a introdução futura de filtros por critérios como género, intervalo de datas ou popularidade, sem necessidade de grandes refatorações.

6 Desafios e Soluções

Durante o desenvolvimento, enfrentámos vários desafios técnicos e conceptuais significativos. Um dos principais problemas foi a inconsistência dos dados na *DBpedia*. Muitos jogos tinham propriedades definidas em namespaces `dbo:` ou `dbp:` mas não em ambos, o que exigiu queries flexíveis usando `UNION` para cobrir ambos os casos.

Outro desafio importante foi o tratamento de plataformas e nomes mal formatados. Frequentemente, os dados vinham como *URIs* pouco legíveis ou strings com caracteres especiais. A nossa solução incluiu o desenvolvimento de funções de normalização para extrair apenas o identificador relevante e apresentá-lo de forma limpa.

A ausência de dados também se revelou um problema comum. Muitos jogos não tinham informação sobre publicador, data de lançamento ou outros campos importantes. Para resolver isto, implementámos um sistema de validação rigoroso para evitar criar instâncias incompletas na ontologia.

Do ponto de vista da performance, a busca de dados da *DBpedia* mostrou ser a parte mais demorada, uma vez que vamos buscar informação dos jogos um a um, algo que (em teoria) poderia ser evitado através do uso de uma única query *SPARQL* para buscar a informação toda de uma vez, porém isso mostrou-se ineficiente na prática.

Finalmente, garantir que a ontologia mantinha coerência semântica com milhares de instâncias exigiu testes constantes e uma abordagem de serialização incremental com a biblioteca `rdflib`, que permite detetar e corrigir problemas à medida que surgiam.

7 Conclusão

O projeto demonstrou a aplicação prática de tecnologias semânticas como *OWL*, *RDF* e *SPARQL* no domínio dos jogos digitais, que resultou numa ontologia funcional e acessível.

A aplicação web criada permite explorar facilmente os dados extraídos e está preparada para futuras expansões ou integrações com outras fontes, como a Wikidata.