



**UNIVERSIDADE FEDERAL DO CEARÁ  
CAMPUS SOBRAL  
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO**

**ALANNA MARIA MACHADO ALVES PAIVA**

**UMA VISÃO GERAL DAS PRÁTICAS DA WEB SEMÂNTICA NO CENÁRIO ATUAL**

**SOBRAL  
2025**

ALANNA MARIA MACHADO ALVES PAIVA

UMA VISÃO GERAL DAS PRÁTICAS DA WEB SEMÂNTICA NO CENÁRIO ATUAL

Trabalho de Conclusão de Curso apresentado  
ao Curso de Graduação em Engenharia de  
Computação do Campus Sobral da Universidade  
Federal do Ceará, como requisito parcial à  
obtenção do grau de bacharel em Engenharia de  
Computação.

Orientador: Prof. Dr. Fischer Jônatas  
Ferreira

SOBRAL

2025

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Sistema de Bibliotecas  
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

M129v Machado Alves Paiva, Alanna Maria.  
UMA VISÃO GERAL DAS PRÁTICAS DA WEB SEMÂNTICA NO CENÁRIO ATUAL / Alanna Maria  
Machado Alves Paiva. – 2025.  
91 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Sobral,  
Curso de Engenharia da Computação, Sobral, 2025.  
Orientação: Prof. Dr. Fischer Jônatas Ferreira.

1. Web Semântica. 2. Ontologias. 3. RDF. 4. SHACL. 5. Tecnologias. I. Título.

CDD 621.39

---

ALANNA MARIA MACHADO ALVES PAIVA

## UMA VISÃO GERAL DAS PRÁTICAS DA WEB SEMÂNTICA NO CENÁRIO ATUAL

Trabalho de Conclusão de Curso apresentado  
ao Curso de Graduação em Engenharia de  
Computação do Campus Sobral da Universidade  
Federal do Ceará, como requisito parcial à  
obtenção do grau de bacharel em Engenharia de  
Computação.

Aprovada em: 29 de Julho de 2025

### BANCA EXAMINADORA

---

Prof. Dr. Fischer Jônatas Ferreira (Orientador)  
Universidade Federal de Itajubá - UNIFEI

---

Prof. Dr. Evilásio Costa júnior  
Universidade Federal do Ceará - UFC

---

Prof. Dr. Eduardo Jabbur Machado  
Universidade Federal de Itajubá - UNIFEI

Dedico todo e qualquer sucesso meu aos meus pais, que, sob muito sol, me fizeram chegar até aqui pela sombra e água fresca. E, principalmente, à minha mãe, que me deu suas asas para que eu pudesse voar mais alto.

## AGRADECIMENTOS

Agradeço primeiramente a Deus, que me deu as últimas forças para conseguir concluir a graduação, a minha família, meus pais Guiomar e Mairton por todo o apoio, dedicação e incentivo, que mesmo diante todas as dificuldades sempre se esforçaram ao máximo para me proporcionar uma vida e uma educação de qualidade, e ao meu irmão caçula Vinícius.

A minha dupla acadêmica, Ananda, agradeço por me acompanhar nessa jornada e ajudar a torná-la mais leve. Dividir todos esses anos com você foi um presente. Aos meus amigos Ailton e Vitor, agradeço por toda a ajuda prestada durante a graduação, sem vocês teria sido bem mais difícil chegar até aqui. A vocês três, minha gratidão por estarem ao meu lado nos estudos, nos trabalhos, nas madrugadas, no dia a dia e na vida.

Ao meu namorado, que me acompanhou nessa reta final, agradeço por sua confiança, apoio, incentivo, amor e esperança, e por acreditar mais em mim do que eu mesma. Você foi meu abrigo.

Ao meu professor orientador, Dr. Fischer Ferreira, agradeço imensamente por toda dedicação, paciência e apoio prestado durante esses anos.

Agradeço também a todas as pessoas que mesmo distante, se fizeram presentes e luz em minha vida de alguma forma, as amigas companheiras de calouradas e cerveja, aos amigos da dança e aos colegas de turma.

Por fim, e não menos importante, agradeço a mim mesma, que, por um acaso do destino, ingressou em um curso e em uma área que não conhecia, mas teve a coragem de permanecer persistente até o fim. Mesmo diante de tantas quedas, sou grata por não ter desistido.

## RESUMO

**CONTEXTO:** A Web Semântica representa uma extensão da Web convencional, visando estruturar e interconectar dados disponíveis na internet para melhor compreensão e processamento por máquinas. No cenário tecnológico atual, continua a ser uma área de pesquisa e aplicação significativa, com desenvolvimentos contínuos e crescente relevância em diversas áreas.

**MOTIVAÇÃO:** A ausência de um estudo abrangente sobre as práticas mais comuns da Web Semântica na literatura é uma lacuna que precisa ser preenchida. Embora haja informações teóricas sobre a Web Semântica, há uma carência de análises práticas, o que limita a compreensão de como essa tecnologia é efetivamente usada. Este estudo visa preencher essa lacuna, identificando e documentando as práticas mais usadas na Web Semântica. Ao fazê-lo, busca-se oferecer *insights* úteis para profissionais e pesquisadores, contribuindo para um melhor entendimento e aplicação prática dessa tecnologia em diferentes contextos.

**OBJETIVO:** Em vista disso, este trabalho tem como propósito oferecer uma visão geral da Web Semântica no contexto atual, realizando uma análise das práticas mais utilizadas nos repositórios do GitHub. Assim, objetiva-se fornecer informações relevantes para a utilização da Web Semântica em diferentes sistemas e aplicações.

**METODOLOGIA:** Este estudo adota uma abordagem metodológica que visa obter uma análise percentual das práticas da Web Semântica. O processo inicia-se com a revisão literária, seguida pela definição de métricas específicas com base nas funcionalidades oferecidas pelos repositórios no GitHub. Posteriormente, realiza-se a seleção criteriosa dos repositórios, levando em consideração esses critérios específicos para identificar projetos relevantes. A pesquisa terá como resultado a documentação das práticas, incluindo uma análise do percentual de projetos que as utilizam, detalhando seu uso e avaliando sua importância.

**BENEFICIADOS:** Os resultados deste estudo beneficiam desenvolvedores, empresas e pesquisadores. Desta forma, eles poderão observar as características levantadas sobre as principais tecnologias da Web Semântica e utilizar em seus projetos. Além disso, servirá como ponto de referência para pesquisadores que atuam na área, contribuindo com o avanço no desenvolvimento da Web Semântica para aplicações Web.

**Palavras-chave:** Web Semântica, W3C, Ontologias, RDF, OWL, SPARQL, SKOS, SHACL, Tecnologias.

## ABSTRACT

**CONTEXT:** The Semantic Web represents an extension of the conventional Web, aiming to structure and interconnect data available on the Internet for better understanding and processing by machines. In today's technological landscape, it continues to be a significant area of research and application, with ongoing developments and growing relevance in various areas. **MOTIVATION:** The absence of a comprehensive study of the most common Semantic Web practices in the literature is a gap that needs to be filled. Although there is theoretical information on the Semantic Web, there is a lack of practical analysis, which limits the understanding of how this technology is actually used. This study aims to fill this gap by identifying and documenting the most commonly used practices on the Semantic Web. In doing so, it aims to provide useful *insights* for professionals and researchers, contributing to a better understanding and practical application of this technology in different contexts. **OBJECTIVE:** The purpose of this paper is to provide an overview of the Semantic Web in the current context by analyzing the most widely used practices in GitHub repositories. The aim is to provide relevant information for using the Semantic Web in different systems and applications. **METHODOLOGY:** This study adopts a methodological approach that aims to obtain a percentage analysis of Semantic Web practices. The process begins with a literature review, followed by the definition of specific metrics based on the functionalities offered by the repositories on GitHub. Subsequently, a careful selection of repositories is carried out, taking into account these specific criteria to identify relevant projects. The research will result in the documentation of practices, including an analysis of the percentage of projects that use them, detailing their use and evaluating their importance. **BENEFITS:** The results of this study will benefit developers, companies and researchers. They will be able to observe the characteristics of the main Semantic Web technologies and use them in their projects. In addition, it will serve as a point of reference for researchers working in the area, contributing to progress in the development of the Semantic Web for web applications.

**Keywords:** Semantic Web, W3C, Ontologies, RDF, OWL, SPARQL, SKOS, SHACL, Technologies.

## LISTA DE FIGURAS

|                                                                                                                            |    |
|----------------------------------------------------------------------------------------------------------------------------|----|
| Figura 1 – Estrutura do <i>HyperText Markup Language</i> (HTML5) . . . . .                                                 | 18 |
| Figura 2 – Assistente de voz em ambiente de saúde personalizado . . . . .                                                  | 22 |
| Figura 3 – Web Semântica em Assistentes Virtuais . . . . .                                                                 | 23 |
| Figura 4 – Representação Gráfica de ontologia para humanos . . . . .                                                       | 29 |
| Figura 5 – Tabela de Ontologia para "Curso de Programação Avançado" . . . . .                                              | 31 |
| Figura 6 – Representação de agentes inteligentes com base em grafos de conhecimento. . . . .                               | 34 |
| Figura 7 – Agente como “Camada Inteligente” sobre web services . . . . .                                                   | 35 |
| Figura 8 – Web Service como fonte de dados para um agente . . . . .                                                        | 36 |
| Figura 9 – Ontologia em serviço na Internet . . . . .                                                                      | 38 |
| Figura 10 – Funcionamento do LinkedData . . . . .                                                                          | 40 |
| Figura 11 – Estrutura <i>Resource Description Framework</i> (RDF) para representação de relações entre conceitos . . . . . | 43 |
| Figura 12 – Analogia de representação do SHACL . . . . .                                                                   | 54 |
| Figura 13 – Etapas metodológicas . . . . .                                                                                 | 59 |
| Figura 14 – Fórmula para o cálculo percentual . . . . .                                                                    | 74 |
| Figura 15 – Processos para a modelagem de filtração . . . . .                                                              | 75 |
| Figura 16 – Percentual de Uso das Práticas da Web Semântica . . . . .                                                      | 78 |

## LISTA DE TABELAS

|                                                                                           |    |
|-------------------------------------------------------------------------------------------|----|
| Tabela 1 – Elementos do padrão Dublin Core . . . . .                                      | 26 |
| Tabela 2 – Comparativo entre <i>Dublin Core</i> e <i>Schema.org</i> . . . . .             | 28 |
| Tabela 3 – Componentes de uma Tríplice RDF . . . . .                                      | 42 |
| Tabela 4 – Recursos e suas características em RDF . . . . .                               | 46 |
| Tabela 5 – Descrição das sub-linguagens do OWL . . . . .                                  | 50 |
| Tabela 6 – Indicadores utilizados para análise de projetos no <i>GitHub</i> . . . . .     | 60 |
| Tabela 7 – Indicadores com limiar mínimo . . . . .                                        | 61 |
| Tabela 8 – Repositórios e tecnologias utilizadas . . . . .                                | 77 |
| Tabela 9 – Distribuição dos projetos segundo o número de tecnologias utilizadas . . . . . | 79 |
| Tabela 10 – Linguagens de programação utilizadas nos repositórios analisados . . . . .    | 80 |
| Tabela 11 – Distribuição dos repositórios por ano da última atualização . . . . .         | 81 |
| Tabela 12 – Popularidade dos repositórios em termos de Stars, Forks e Watchers . . . . .  | 82 |
| Tabela 13 – Top 5 repositórios mais populares e suas respectivas tecnologias . . . . .    | 83 |

## LISTA DE ABREVIATURAS E SIGLAS

|       |                                          |
|-------|------------------------------------------|
| API   | <i>Application Programming Interface</i> |
| HTML  | <i>HyperText Markup Language</i>         |
| HTTP  | <i>Hypertext Transfer Protocol</i>       |
| JSON  | <i>JavaScript Object Notation</i>        |
| OWL   | <i>Web Ontology Language</i>             |
| RDF   | <i>Resource Description Framework</i>    |
| SHACL | <i>Shapes Constraint Language</i>        |
| URI   | <i>Uniform Resource Identifier</i>       |
| URIs  | <i>Uniform Resource Identifiers</i>      |
| URL   | <i>Uniform Resource Locator</i>          |
| W3C   | <i>World Wide Web Consortium</i>         |
| WWW   | <i>World Wide Web</i>                    |
| XML   | <i>Extensible Markup Language</i>        |

## SUMÁRIO

|       |                                                                                      |    |
|-------|--------------------------------------------------------------------------------------|----|
| 1     | <b>INTRODUÇÃO</b>                                                                    | 13 |
| 2     | <b>FUNDAMENTOS</b>                                                                   | 15 |
| 2.1   | <b>World Wide Web (W3)</b>                                                           | 15 |
| 2.1.1 | <i>Hypertext Markup Language (HTML)</i>                                              | 16 |
| 2.1.2 | <i>W3C</i>                                                                           | 20 |
| 2.1.3 | <i>A Evolução da Web</i>                                                             | 20 |
| 2.2   | <b>Web Semântica</b>                                                                 | 21 |
| 2.2.1 | <i>Metadados</i>                                                                     | 24 |
| 2.2.2 | <i>Ontologias</i>                                                                    | 28 |
| 2.2.3 | <i>Agentes Inteligentes</i>                                                          | 33 |
| 2.2.4 | <i>Web Services</i>                                                                  | 34 |
| 2.2.5 | <i>Linked Data</i>                                                                   | 38 |
| 2.3   | <b>Ferramentas e Tecnologias da Web Semântica</b>                                    | 40 |
| 2.3.1 | <i>Resource Description Framework (RDF)</i>                                          | 41 |
| 2.3.2 | <i>RDF Schema (RDFS)</i>                                                             | 45 |
| 2.3.3 | <i>JavaScript Object Notation for Linked Data (JSON-LD)</i>                          | 47 |
| 2.3.4 | <i>Extensible Markup Language (XML)</i>                                              | 48 |
| 2.3.5 | <i>Web Ontology Language (OWL)</i>                                                   | 48 |
| 2.3.6 | <i>Simple Knowledge Organization System (SKOS)</i>                                   | 51 |
| 2.3.7 | <i>Shapes Constraint Language (SHACL)</i>                                            | 53 |
| 2.3.8 | <i>Protocol and Resource Description Framework Query Language (SPARQL)</i>           | 56 |
| 2.4   | <b>Web Semântica no cenário atual</b>                                                | 57 |
| 3     | <b>METODOLOGIA</b>                                                                   | 59 |
| 3.1   | <b>Levantamento das práticas descritas na literatura relacionada à Web Semântica</b> | 59 |
| 3.2   | <b>Definição de critérios para a busca dos sistemas alvo no <i>GitHub</i>.</b>       | 60 |
| 3.3   | <b>Seleção dos projetos do <i>GitHub</i> que estão inseridos nesse critério</b>      | 61 |
| 3.3.1 | <i>Automatização da seleção dos projetos</i>                                         | 62 |
| 3.4   | <b>Mineração das práticas da web semântica utilizadas por esses sistemas</b>         | 68 |
| 3.4.1 | <i>Filtragem por linguagens</i>                                                      | 69 |

|              |                                                                                                  |    |
|--------------|--------------------------------------------------------------------------------------------------|----|
| <b>3.4.2</b> | <i>Filtragem por extensões de arquivos</i>                                                       | 70 |
| <b>3.4.3</b> | <i>Coleta e consolidação dos dados dos repositórios</i>                                          | 71 |
| <b>3.5</b>   | <b>Análise e documentação do percentual de práticas encontradas</b>                              | 72 |
| <b>3.5.1</b> | <i>Análise quantitativa das práticas da Web Semântica</i>                                        | 73 |
| <b>3.6</b>   | <b>Perguntas de Pesquisa</b>                                                                     | 74 |
| <b>4</b>     | <b>RESULTADOS E DISCUSSÕES</b>                                                                   | 75 |
| <b>4.1</b>   | <b>Apresentação Geral busca dos projetos</b>                                                     | 75 |
| <b>4.2</b>   | <b>Análise Quantitativa</b>                                                                      | 76 |
| <b>4.2.1</b> | <i>Frequência de adoção das tecnologias da Web Semântica</i>                                     | 76 |
| <b>4.2.2</b> | <i>Características dos Projetos e Linguagens utilizadas</i>                                      | 80 |
| <b>4.2.3</b> | <i>Atividade Recente dos Repositórios</i>                                                        | 81 |
| <b>4.2.4</b> | <i>Análise da Adoção Tecnológica nos repositórios mais populares e Engajamento da Comunidade</i> | 81 |
| <b>4.3</b>   | <b>Lições Aprendidas</b>                                                                         | 83 |
| <b>5</b>     | <b>CONCLUSÕES E TRABALHOS FUTUROS</b>                                                            | 85 |
|              | <b>REFERÊNCIAS</b>                                                                               | 87 |

## 1 INTRODUÇÃO

A evolução da tecnologia tem transformado significativamente a maneira como as pessoas interagem, consomem e compartilham informações, tornando a Web um ambiente cada vez mais dinâmico, conectado e essencial no cotidiano. Nesse cenário, a *World Wide Web* (WWW) consolidou-se como uma das maiores inovações tecnológicas do século XX, ao possibilitar a disseminação global de informações e o acesso democratizado ao conhecimento. Contudo, à medida que o volume de dados na Web cresce de forma exponencial, a Web tradicional enfrenta limitações para oferecer respostas precisas, integradas e personalizadas às necessidades dos usuários, devido à falta de semântica nos dados.

Para superar essas limitações, Tim Berners-Lee, criador da Web, propôs, no início dos anos 2000, o conceito de Web Semântica, uma evolução da Web que visa tornar os dados publicados mais estruturados, interconectados e comprehensíveis para as máquinas, promovendo uma Web não apenas de documentos legíveis por humanos, mas também de dados interpretáveis por agentes automatizados (BERNERS-LEE *et al.*, 2001). Tal abordagem busca superar desafios clássicos dos sistemas de recuperação de informação, como a integração de fontes heterogêneas, o refinamento de buscas via ontologias e a filtragem segundo as preferências do usuário, além de aprimorar as interfaces e a experiência de navegação (HITZLER, 2021). Nesse cenário, a Web Semântica se consolida como um campo promissor para o desenvolvimento de soluções mais inteligentes, interoperáveis e alinhadas às necessidades contemporâneas de gestão de dados.

A motivação central deste trabalho reside na necessidade de compreender como as tecnologias da Web Semântica vêm sendo efetivamente aplicadas em sistemas reais. Apesar de amplamente abordadas na literatura, ainda são escassas as evidências empíricas sobre sua presença e uso em projetos de código aberto. Assim, este trabalho tem como objetivo investigar a adoção dessas tecnologias no cenário atual em projetos hospedados no *GitHub*, analisando sua frequência de uso, características técnicas, padrões de atualização e engajamento comunitário. Busca-se, dessa forma, não apenas quantificar a presença dessas tecnologias, mas também identificar como elas são aplicadas e quais tendências emergem a partir dessa análise.

Desta forma, este trabalho foi composto de algumas etapas para a obtenção dos resultados. Primeiramente, realizou-se um levantamento teórico sobre as tecnologias da Web Semântica, com base na literatura especializada e documentos técnicos, a fim de compreender os conceitos fundamentais e identificar as principais ferramentas, tecnologia e padrões utilizados. Em seguida, definiu-se um conjunto abrangente de critérios para a seleção dos repositórios a

serem analisados, de modo a garantir a relevância e a atividade recente dos projetos escolhidos. A partir dos repositórios selecionados, realizou-se a mineração das práticas da Web Semântica presentes em cada projeto, verificando a adoção efetiva das principais tecnologias e a frequência de uso. Por fim, os dados coletados foram submetidos a uma análise quantitativa, possibilitando identificar padrões de adoção tecnológica, níveis de popularidade e engajamento da comunidade, bem como o percentual de utilização de cada prática nos projetos.

Os resultados obtidos a partir dessa análise oferecem uma visão geral sobre a presença e a aplicação das tecnologias da Web Semântica em projetos de código aberto, permitindo identificar quais são as mais utilizadas, de que forma são implementadas e em que medida os projetos se mantêm ativos e relevantes. Além disso, revelam padrões de popularidade e participação da comunidade em torno dessas tecnologias, contribuindo para uma compreensão mais clara de sua maturidade e impacto no ecossistema atual.

Este trabalho está organizado seguindo a divisão por capítulos. No Capítulo 2 aborda a fundamentação teórica deste estudo, descrevendo os conceitos e tecnologias da Web Semântica. Já o Capítulo 3 descreve a metodologia empregada na condução deste trabalho. No Capítulo 4 são apresentados os resultados encontrados e a discussão dos mesmos. Por fim, no Capítulo 5, são apresentadas as considerações finais.

## 2 FUNDAMENTOS

Neste capítulo são apresentados os fundamentos que serviram de base para a elaboração e construção deste trabalho, e um breve histórico das principais transformações causadas pela tecnologia na maneira pela qual as pessoas interagem e se relacionam. Inicialmente, será apresentada uma visão geral da Rede de Alcance Geral, WWW, expondo seu contexto histórico e suas principais características. Por fim, apresenta-se de forma abrangente sobre a Web Semântica e suas ferramentas, linguagens e aplicações, destacando algumas áreas que podem ser beneficiadas com as suas tecnologias.

### 2.1 World Wide Web (W3)

A World Wide Web, ou simplesmente Web, é um dos principais componentes da internet, responsável por viabilizar o acesso e o compartilhamento de informações em formato de documentos multimídia, como textos, imagens e vídeos, por meio de navegadores. Criada por Tim Berners-Lee no início dos anos 1990, a Web surgiu com o objetivo de permitir o compartilhamento autorizado de informações entre cientistas e instituições de pesquisa em diferentes partes do mundo (BERNERS-LEE, 1999).

A estrutura da Web baseia-se no conceito de hiperlinks que são conexões que interligam páginas e recursos distintos. Por meio desses links, os usuários podem navegar de forma não linear entre conteúdos diversos, explorando uma ampla rede de informações. Cada recurso disponível na Web é identificado por um endereço único, chamado URL (Uniform Resource Locator), que possibilita sua localização e acesso específicos (BERNERS-LEE, 1992).

Berners-Lee também foi responsável pelo desenvolvimento do protocolo HTTP (Hypertext Transfer Protocol), que estabelece a comunicação entre servidores e clientes de forma leve e eficiente. Além disso, ele introduziu a linguagem HTML (Hypertext Markup Language), utilizada para estruturar os conteúdos exibidos nas páginas web. Com esses componentes (URL, HTTP e HTML) formou-se a base tecnológica que tornou a Web acessível e funcional em escala global.

O projeto da World Wide Web foi concebido com a visão de construir um espaço universal de informações, no qual cada item pudesse ser recuperado a partir de uma referência única. Desde então, a Web transformou profundamente a forma como as pessoas acessam e compartilham informações, democratizando o conhecimento e facilitando a comunicação

em escala planetária. Tornou-se uma plataforma essencial para diversas atividades cotidianas, como pesquisa, comunicação, educação, entretenimento e comércio eletrônico. Sua evolução contínua segue moldando a maneira como indivíduos, organizações e sistemas interagem na era digital (AGARWAL; SASTRY, 2022; GOYAL *et al.*, 2023; CHAQFEH *et al.*, 2023).

### 2.1.1 *Hypertext Markup Language (HTML)*

A Web, em sua concepção original, não foi desenvolvida com foco na semântica. Contudo, com o crescimento acelerado da informação online e a necessidade de melhorar a organização, a acessibilidade e a indexação dos conteúdos, tornou-se necessário adotar mecanismos que conferissem significado estrutural aos elementos das páginas. O HTML (Hypertext Markup Language) surgiu como uma linguagem padrão para atender a essa demanda, oferecendo uma base comum para a criação e interconexão de documentos hipertextuais. Projetado para ser simples e universal, o HTML permitiu a construção da estrutura essencial das primeiras páginas da Web. Por ser uma linguagem de marcação, sua principal função era definir a organização do conteúdo, utilizando elementos básicos como cabeçalhos, listas, parágrafos e divisões.

O Listing 2.1.1 apresenta um exemplo de código em HTML na versão 4. A estrutura é iniciada pela diretiva `<!DOCTYPE>` (linha 1), que declara o tipo de documento e a versão da linguagem utilizada. Essa declaração passou por ajustes ao longo das versões do HTML para acompanhar as novas especificações. No corpo do código, diferentes elementos desempenham funções estruturais. No exemplo apresentado, as tags `<div>` são utilizadas para dividir a página em seções, mas não atribuem significado semântico ao conteúdo que encapsulam (BERNERS-LEE *et al.*, 1992).

- `<head>` (linha 3): contém metadados do documento, incluindo título, codificação e outras informações não exibidas diretamente na página;
- `<title>` (linha 4): define o título que será exibido na aba do navegador;
- `<body>` (linha 6): envolve todo o conteúdo visível da página.

```

1  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
2  <html>
3  <head>
4      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5      <title>Exemplo de Página HTML</title>
6  </head>
7  <body>
8      <div id="header">Bem-vindo à Página HTML</div>
9      <p>Esta é uma página de exemplo criada em HTML.</p>
10
11     <div id="menu">
12         <div class="post">Início</div>
13     </div>
14
15     <div id="footer">Rodapé da página</div>
16 </body>
17 </html>

```

Listing 2.1.1 – Exemplo de página HTML

O HTML4<sup>1</sup> apresenta uma estrutura simples e funcional, mas sua limitação na ausência de elementos com valor semântico explícito. Isso dificultava a interpretação automatizada do conteúdo, especialmente por motores de busca e tecnologias assistivas. Com o tempo, a linguagem evoluiu, incorporando elementos capazes de expressar melhor a função de cada parte da página. Essa evolução resultou no HTML5<sup>2</sup>, versão mais recente e amplamente adotada.

A Web Semântica surgiu justamente da necessidade de tornar o conteúdo mais inteligível tanto para humanos quanto para máquinas. No HTML4, como ilustrado no Listing 2.1.1, o uso predominante de elementos genéricos como `<div>` comprometia a compreensão do propósito do conteúdo. Embora tais elementos organizassem visualmente a página, não informavam à máquina se abrigavam um menu, um rodapé ou um artigo, dificultando a extração de significado.

A introdução do HTML5 representou um avanço significativo nesse aspecto. O HTML5, embora não seja tecnicamente parte dos padrões de Web Semântica definidos pelo W3C (como RDF<sup>3</sup>, OWL<sup>4</sup> e SPARQL<sup>5</sup>), é um facilitador da semântica no nível estrutural da Web. Essa versão do HTML passou a oferecer elementos com valor semântico explícito, como `<header>`, `<footer>`, `<article>`, `<section>` e `<nav>` (PILGRIM, 2010). Com essas novas tags, o desenvolvedor consegue estruturar o conteúdo de forma mais expressiva e compreensível

<sup>1</sup> <<https://www.w3.org/TR/html4/>>

<sup>2</sup> <<https://html.spec.whatwg.org/multipage/>>

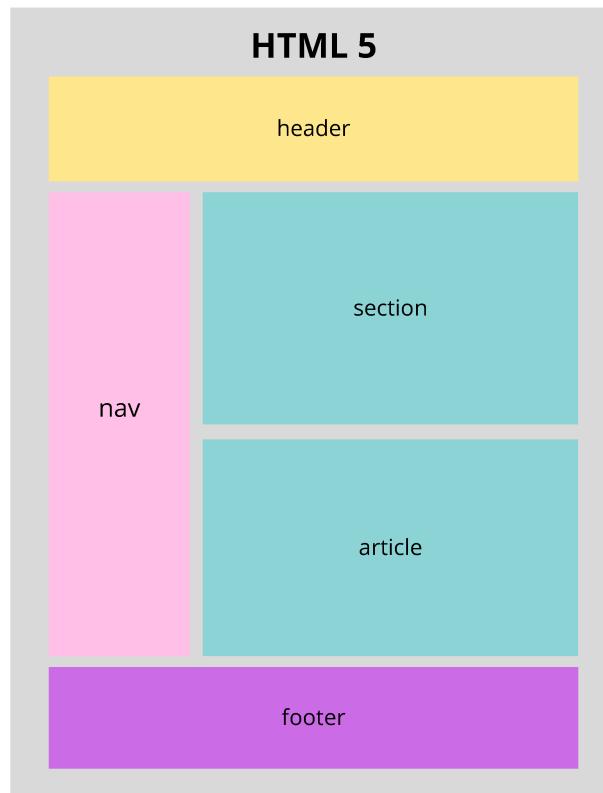
<sup>3</sup> <<https://www.w3.org/TR/rdf11-concepts/>>

<sup>4</sup> <<https://www.w3.org/TR/owl2-overview/>>

<sup>5</sup> <<https://www.w3.org/TR/sparql11-query/>>

para diferentes agentes da Web, como navegadores, mecanismos de busca e leitores de tela. A Figura 1 ilustra a estrutura típica de uma página em HTML5, com ênfase na distribuição semântica dos elementos (LAWSON; SHARP, 2011).

Figura 1 – Estrutura do HTML5



Fonte: Produzido pela autora.

Para exemplificar a aplicação dos elementos semânticos, o Listing 2.1.2 apresenta uma estrutura básica de página em HTML5. Nesse exemplo, observa-se a substituição de `<div>` genéricos por elementos com significado próprio, melhorando a compreensão do conteúdo por parte das máquinas.

- `<body>` (linha 8): abriga todo o conteúdo da página, agora estruturado com elementos semânticos;
- `<header>` (linha 9): representa o cabeçalho da página ou seção, podendo conter títulos, logotipos e informações introdutórias;
- `<nav>` (linha 12): identifica a área de navegação, contendo links para outras seções ou páginas;
- `<article>` (linha 15): delimita um conteúdo independente e autocontido, como uma notícia ou postagem;
- `<section>` (linha 16): agrupa conteúdo tematicamente relacionado dentro de um artigo

ou página;

- <footer> (linha 20): define o rodapé, geralmente contendo informações auxiliares como contato ou direitos autorais.

```

1      <!DOCTYPE html>
2      <html lang="en">
3          <head>
4              <meta charset="UTF-8" />
5              <meta name="viewport" content="width=device-width,
6                  initial-scale=1.0" />
7              <title>Exemplo de Página HTML</title>
8          </head>
9          <body>
10             <header class="header" id="header">
11                 Cabeçalho da página
12             </header>
13             <nav class="nav">
14                 Menu
15             </nav>
16             <article>
17                 <section id="inicio">
18                     <h2>Bem-vindo ao Meu Site!</h2>
19                 </section>
20             </article>
21             <footer>
22                 <p>Rodapé da página</p>
23             </footer>
24         </body>
25     </html>

```

Listing 2.1.2 – Exemplo de página HTML5

O exemplo evidencia como o HTML5 promove uma estruturação mais clara e informativa, favorecendo a interoperabilidade entre aplicações web. A adoção de elementos semânticos fortalece a capacidade de agentes automatizados em interpretar, classificar e reutilizar o conteúdo, além de melhorar significativamente a experiência de navegação para usuários que dependem de tecnologias assistivas. Essa transição é um dos pilares fundamentais da Web Semântica, ao permitir que o conteúdo seja não apenas exibido, mas compreendido. Vale destacar que, embora o HTML5 tenha introduzido elementos semânticos importantes, ele atua principalmente na camada de estrutura da informação. A Web Semântica, conforme definida pelo W3C, envolve também a representação formal de conhecimento por meio de tecnologias como RDF, OWL e SPARQL.

### 2.1.2 W3C

O *World Wide Web Consortium* (W3C)<sup>6</sup> é um consórcio internacional responsável pelo desenvolvimento dos padrões que orientam o funcionamento da Web. Fundado por Tim Berners-Lee, o W3C tem como principal missão promover uma arquitetura técnica consistente, capaz de acompanhar a rápida evolução das tecnologias web e garantir sua interoperabilidade (CONSORTIUM, 2023).

Os padrões definidos pelo W3C servem como referência para o desenvolvimento de sites, navegadores e dispositivos, assegurando que todos possam acessar os recursos da Web de forma eficiente e compatível. Além disso, esses padrões oferecem uma base sólida para enfrentar desafios fundamentais, como acessibilidade, internacionalização, privacidade e segurança.

A criação do W3C também teve um papel estratégico na coordenação do desenvolvimento da Web. Diante do avanço acelerado das tecnologias, tornou-se necessário reunir pesquisadores, desenvolvedores e empresas em torno de um esforço comum, evitando fragmentações e promovendo a adoção de diretrizes unificadas (BERNERS-LEE *et al.*, 1992; CONSORTIUM, 2023).

### 2.1.3 A Evolução da Web

A Web passou por transformações significativas desde sua criação. Originalmente, a estrutura da chamada Web tradicional (ou Web 1.0) foi projetada apenas para apresentar informações, sem mecanismos que permitissem sua interpretação por computadores. Nesse modelo, os conteúdos eram processados exclusivamente por seres humanos, o que limitava a automação e o uso inteligente das informações (BREITMAN, 2005).

Essa limitação inspirou o desenvolvimento da Web Semântica, uma abordagem projetada para expandir a estrutura tradicional das páginas web. Seu objetivo é incorporar metadados que descrevem não apenas as páginas, mas também suas relações com outras entidades. Tais metadados fornecem informações estruturadas, permitindo que agentes automatizados naveguem pela Web e executem tarefas que, anteriormente, só poderiam ser realizadas por seres humanos devido às restrições da Web tradicional, conforme discutido por Segaran *et al.* (2009). Em essência, a Web Semântica busca superar as limitações da Web convencional, acrescentando uma camada adicional de significado e compreensão às informações online.

---

<sup>6</sup> <<https://www.w3.org/>>

Com essa abordagem, a Web deixa de ser apenas um repositório de dados e passa a funcionar como uma rede de informações interconectadas e compreensíveis por sistemas computacionais. Isso amplia as possibilidades de automação, personalização e integração entre aplicações, tornando a Web mais eficiente e acessível para diferentes públicos e dispositivos (AGARWAL; SASTRY, 2022; GOYAL *et al.*, 2023; CHAQFEH *et al.*, 2023).

## 2.2 Web Semântica

O conceito de Web Semântica foi introduzido em 2001 por Berners-Lee *et al.* (2001), o inventor da Web, com o objetivo de propor uma nova forma de estruturar e interligar informações na internet. Diferentemente da Web tradicional, voltada basicamente para usuários humanos, a Web Semântica por meio de um conjunto de ferramentas, padrões e tecnologias possibilitam que máquinas possam processar, interpretar e realizar inferências sobre dados estruturados publicados na Web (EIS, 2017). Isso permite que as máquinas auxiliem os usuários na obtenção de respostas mais precisas e relevantes, reduzindo a necessidade de intervenção humana, como ocorre nos mecanismos de busca convencionais, a exemplo do Google.

Para introduzir o tema de forma prática, os Cenários das Figuras 2 e 3 apresentam duas situações reais de aplicação da Web Semântica no cotidiano de duas pessoas (Juliana e Ana), evidenciando seu potencial em diferentes contextos.

O Cenário da Figura 2 descreve o uso de um assistente virtual para realizar uma busca por restaurantes, utilizando tecnologias como RDF, SPARQL e vocabulários padronizados (por exemplo, schema.org em JSON-LD<sup>7</sup>). Essa tecnologia permite transformar dados desconexos em informações organizadas e úteis, respondendo de forma contextualizada e precisa à necessidade do usuário.

---

<sup>7</sup> <<https://www.w3.org/TR/json-ld11/>>

### Cenário 1 — Assistente de voz em ambiente de saúde personalizado

Juliana está organizando um jantar com amigos e pergunta ao seu assistente virtual: "*Qual o restaurante japonês mais bem avaliado aberto agora perto de mim?*"

O assistente virtual, que utiliza tecnologias da Web Semântica, executa os seguintes passos:

1. Consulta fontes estruturadas como o Google Knowledge Graph, que utiliza dados em RDF extraídos de sites anotados com JSON-LD e schema.org.
2. Identifica restaurantes classificados como schema:Restaurant com schema:cuisine igual a "Japanese".
3. Verifica a geolocalização do usuário, filtra por restaurantes em um raio de 5 km e abertos no momento.
4. Ordena os resultados com base na propriedade schema:aggregateRating.
5. Responde: "*O restaurante Sakura Sushi, com avaliação 4.7 estrelas, está aberto agora e fica a 1.2 km de você. Deseja ligar ou fazer uma reserva?*"

Figura 2 – Assistente de voz em ambiente de saúde personalizado

Já o Cenário da Figura 3 apresenta uma aplicação voltada à área da saúde. Nele, uma paciente com diabetes tipo 2 recebe recomendações personalizadas de um assistente virtual semântico. O sistema é capaz de analisar ontologias médicas, integrar dados de múltiplas fontes, sugerir ações clínicas e facilitar a comunicação entre paciente e profissionais de saúde. Essa abordagem demonstra o potencial da Web Semântica para transformar dados de saúde em decisões inteligentes e contextualizadas.

### Exemplo Atual — Web Semântica em Assistentes Virtuais

Ana, uma paciente com diabetes tipo 2, que utiliza um aplicativo de assistente virtual em seu *smartphone*. Esse aplicativo, integrado com tecnologias de Web Semântica, é capaz de: O assistente virtual, que utiliza tecnologias da Web Semântica, executa os seguintes passos:

1. **Analizar ontologias médicas** (como SNOMED<sup>a</sup>) para interpretar corretamente os registros eletrônicos de saúde de Ana, incluindo exames laboratoriais recentes e histórico de prescrições.
2. **Integrar dados de diferentes fontes**: o assistente acessa dados de dispositivos vestíveis (como um relógio inteligente que mede glicose e frequência cardíaca), registros médicos armazenados em hospitais e dados de farmácias locais.
3. **Fazer recomendações personalizadas**: com base nas diretrizes clínicas extraídas de bases semânticas e em seu histórico, o sistema sugere que Ana faça um exame de hemoglobina glicada, pois detectou uma tendência de elevação nos níveis de glicose nas últimas semanas.
4. **Agendar automaticamente uma consulta**: o assistente usa dados vinculados (Linked Data) para encontrar endocrinologistas próximos, filtrar aqueles que aceitam o plano de saúde de Ana e verificar disponibilidade em sua agenda.
5. **Gerar relatórios comprehensíveis para médicos e pacientes**: as informações são apresentadas usando vocabulários padronizados, facilitando tanto a comunicação entre profissionais quanto a compreensão por parte de Ana.

<sup>a</sup> <<https://www.snomed.org/what-is-snomed-ct>>

Figura 3 – Web Semântica em Assistentes Virtuais

Esses exemplos evidenciam como a Web Semântica está presente em tecnologias contemporâneas, promovendo a integração entre dados e a geração de informações inteligentes, contextualizadas e úteis para os usuários. Seja em atividades cotidianas ou em setores estratégicos como a saúde, essas tecnologias ampliam as possibilidades de personalização e automação, contribuindo para uma Web mais conectada e orientada ao significado.

Os dois cenários ilustram como a Web Semântica pode transformar tarefas cotidianas ao permitir que os computadores compreendam o significado das informações e colaboremativamente com os usuários. Para tornar esses cenários possíveis, é necessário desenvolver uma infraestrutura capaz de organizar e inter relacionar os dados de forma comprehensível para as máquinas. Nesse contexto, surgem os modelos denominados ontologias, que organizam o conhecimento e descrevem as relações entre conceitos, funcionando como um dicionário ou mapa semântico.

Além disso, para compreender plenamente os fundamentos e a aplicação da Web Semântica, é importante considerar alguns temas centrais que compõem sua estrutura e funciona-

mento, tais como: Metadados, Ontologias, Agentes, *Web Services*, Linguagens e Ferramentas.

### 2.2.1 *Metadados*

Metadados são informações que descrevem outros dados. Eles têm papel central na Web Semântica, pois tornam possível organizar, localizar, descrever e gerenciar conteúdos de forma estruturada e compreensível por máquinas. Ao agregar contexto e significado aos recursos da Web, os metadados permitem que sistemas automatizados interpretem o conteúdo das páginas com maior precisão, facilitando sua indexação, classificação e reutilização (ŘEZNÍK *et al.*, 2022).

No contexto da Web, o desenvolvimento e uso de metadados envolve três aspectos principais. O primeiro é a *descrição dos recursos*, ou seja, a definição clara das informações que serão tratadas como metadados. O segundo refere-se à *produção desses metadados*, que, devido ao grande volume de dados disponíveis online, tende a ser realizada por meio de processos automatizados. A criação manual, embora mais precisa em alguns contextos, é inviável em larga escala. Por fim, há o *uso dos metadados*, que envolve sua leitura e processamento por sistemas computacionais, garantindo que o significado semântico original seja preservado durante a manipulação dos dados (DIAS; SANTOS, 2003).

Além desses aspectos, outras questões devem ser consideradas. A diversidade de padrões existentes permite que um mesmo recurso seja descrito por diferentes conjuntos de qualificadores. Por isso, é comum a adoção de padrões específicos para distintos tipos de recursos, como textos acadêmicos, imagens ou produtos comerciais. Outro ponto relevante é a *internacionalização dos metadados*, uma vez que muitos qualificadores foram originalmente definidos em inglês, exigindo adaptações para uso em contextos multilíngues. A geração dos metadados pode ocorrer no momento em que o recurso é criado ou posteriormente, sendo necessária sua atualização sempre que o recurso sofre modificações. Em certos casos, como em resenhas ou comentários de usuários, os metadados podem ser gerados de forma independente. Vale ressaltar ainda que metadados são, por definição, dados, e por isso enfrentam os mesmos desafios relacionados a armazenamento, versionamento, acesso e interpretação (IANNELLA; WAUGH, 1997).

A ideia de criar um padrão universal de metadados que conteplete todas as áreas do conhecimento é frequentemente discutida, mas ainda distante de ser viável. Isso se deve à necessidade de descrever domínios muito diversos, cada um com suas terminologias, estruturas e

requisitos específicos. Uma padronização única implicaria perda de expressividade e exigiria alto grau de especialização para garantir uma descrição eficaz dos recursos (DIAS; SANTOS, 2003).

Diante dessa diversidade, surgiram padrões de metadados com finalidades distintas. Um dos mais amplamente utilizados na Web é o *Dublin Core*<sup>8</sup>, desenvolvido com apoio do W3C. Projetado para ser simples e aplicável a diferentes tipos de documentos eletrônicos, o *Dublin Core* permite a criação e compartilhamento de metadados de forma padronizada, sem exigir conhecimentos técnicos avançados. Ele é composto por 15 elementos principais, como título, autor, data, formato e descrição, sendo amplamente adotado em bibliotecas digitais, repositórios institucionais e catálogos acadêmicos (DZIEKANIAK; KIRINUS, 2004).

Cada elemento do *Dublin Core* possui três propriedades fundamentais: o nome (que o identifica), o atributo (forma como é representado nos metadados) e a definição (descrição do conceito). Por exemplo, em uma fotografia digital, os metadados podem incluir informações como data e hora da captura, modelo do dispositivo, localização geográfica e configurações da câmera. Esses dados tornam o recurso mais facilmente pesquisável, classificável e comprehensível tanto para usuários quanto para sistemas automatizados.

A Tabela 1 apresenta os 15 elementos do padrão *Dublin Core*, com seus respectivos atributos e descrições resumidas. Esses elementos podem ser utilizados de forma flexível para descrever diversos tipos de recursos digitais (Dublin Core Metadata Initiative (DCMI), 2020).

---

<sup>8</sup> <<https://www.dublincore.org/specifications/dublin-core/dces/>>

Tabela 1 – Elementos do padrão Dublin Core

| Elemento    | Atributo    | Descrição                                                |
|-------------|-------------|----------------------------------------------------------|
| Title       | title       | Título do recurso.                                       |
| Creator     | creator     | Pessoa ou entidade responsável pela criação do conteúdo. |
| Subject     | subject     | Tema ou assunto tratado no recurso.                      |
| Description | description | Resumo ou explicação sobre o conteúdo.                   |
| Publisher   | publisher   | Entidade responsável pela disponibilização do recurso.   |
| Contributor | contributor | Colaboradores que participaram da criação do recurso.    |
| Date        | date        | Data associada ao recurso (criação, publicação etc.).    |
| Type        | type        | Categoria ou tipo do recurso (ex: imagem, texto, vídeo). |
| Format      | format      | Formato físico ou digital (ex: PDF, JPEG, HTML).         |
| Identifier  | identifier  | Identificador exclusivo do recurso, como uma URL ou DOI. |
| Source      | source      | Recurso original do qual este deriva.                    |
| Language    | language    | Idioma do conteúdo (ex: pt-BR, en-US).                   |
| Relation    | relation    | Relacionamentos com outros recursos.                     |
| Coverage    | coverage    | Escopo geográfico ou temporal do conteúdo.               |
| Rights      | rights      | Informações sobre direitos autorais e uso do recurso.    |

Embora o *Dublin Core* continue sendo amplamente utilizado em ambientes acadêmicos e documentais, os metadados na Web moderna têm sido frequentemente implementados com o uso de *JSON-LD* (JavaScript Object Notation for Linked Data) (Schema.org, 2024), em conjunto com vocabulários padronizados como o *Schema.org*. Essa abordagem tornou-se o método preferencial de marcação semântica em páginas web, por ser compatível com mecanismos de busca, interoperável com sistemas diversos e não interferir na apresentação visual do conteúdo. O *Schema.org*, mantido por organizações como Google, Microsoft e W3C, oferece uma ampla gama de tipos e propriedades para descrever recursos como artigos, pessoas, eventos, produtos e instituições.

O Listing 2.2.1 apresenta um exemplo de metadados estruturados utilizando *JSON-LD* para descrever um artigo acadêmico.

```

1 <script type="application/ld+json">
2 {
3   "@context": "https://schema.org",
4   "@type": "ScholarlyArticle",
5   "headline": "A Evolução da Web Semântica",
6   "author": {
7     "@type": "Person",
8     "name": "Alanna Maria",
9     "affiliation": {
10       "@type": "Organization",
11       "name": "Universidade Federal do Ceará"
12     }
13   },
14   "publisher": {
15     "@type": "Organization",
16     "name": "Universidade Federal do Ceará"
17   },
18   "datePublished": "2025-07-31",
19   "inLanguage": "pt-BR",
20   "description": "Artigo introdutório sobre o uso de metadados e o padrão
21   → Dublin Core na Web Semântica.",
22   "keywords": ["Web Semântica", "Metadados", "Dublin Core", "Schema.org"],
23   "url": "https://www.famart.edu.br/artigos/web-semantica"
24 }
</script>

```

Listing 2.2.1 – Exemplo de metadados no formato JSON-LD

A Tabela 2 resume as principais diferenças entre o padrão *Dublin Core* e o vocabulário *Schema.org*, evidenciando como cada um atende a contextos específicos. Enquanto o *Dublin Core* se mantém como uma opção simples e eficaz em ambientes institucionais e acadêmicos, o *Schema.org*, especialmente em conjunto com *JSON-LD*, consolidou-se como padrão para aplicações web voltadas à visibilidade e interoperabilidade com sistemas de busca e agentes automatizados.

Tabela 2 – Comparativo entre *Dublin Core* e *Schema.org*

| Critério                             | Dublin Core                                              | Schema.org (com JSON-LD)                                                             |
|--------------------------------------|----------------------------------------------------------|--------------------------------------------------------------------------------------|
| Origem                               | Desenvolvido por DCMI (com apoio do W3C)                 | Criado por Google, Microsoft, Yahoo e Yandex; mantido com apoio do W3C               |
| Objetivo                             | Descrição genérica de recursos digitais                  | Marcação semântica para interpretação automática por motores de busca e sistemas web |
| Formato típico                       | Metatags em HTML (<meta name="dc.title" ...>)            | JSON-LD (bloco <script type="application/ld+json">)                                  |
| Número de elementos                  | 15 elementos principais (ex: title, creator, date)       | Centenas de tipos e propriedades específicas (ex: Person, Article, Event)            |
| Complexidade                         | Simples, de uso genérico                                 | Mais detalhado e contextualizado, requer vocabulário mais amplo                      |
| Facilidade de uso                    | Fácil, especialmente para iniciantes                     | Requer familiaridade com JSON-LD e vocabulários semânticos                           |
| Compatibilidade com motores de busca | Limitada                                                 | Altamente compatível e recomendada por Google e Bing                                 |
| Principais usos                      | Repositórios digitais, bibliotecas, catálogos acadêmicos | SEO, sites comerciais, artigos, produtos, eventos, currículos, instituições          |
| Exemplo de aplicação                 | DSpace, bibliotecas, arquivos públicos                   | Google Scholar, sites institucionais, comércio eletrônico                            |

Em síntese, os metadados desempenham um papel estratégico na estruturação e evolução da Web Semântica (ŘEZNÍK *et al.*, 2022). Ao fornecer uma camada adicional de significado aos conteúdos digitais, eles promovem a interoperabilidade entre sistemas, aprimoram a busca e a recuperação da informação, e tornam possível o desenvolvimento de aplicações mais inteligentes e sensíveis ao contexto. A escolha do padrão de metadados mais adequado depende do tipo de recurso, do público-alvo e dos objetivos da aplicação, sendo o *Dublin Core* e o *Schema.org* duas das soluções mais consolidadas e complementares nesse cenário.

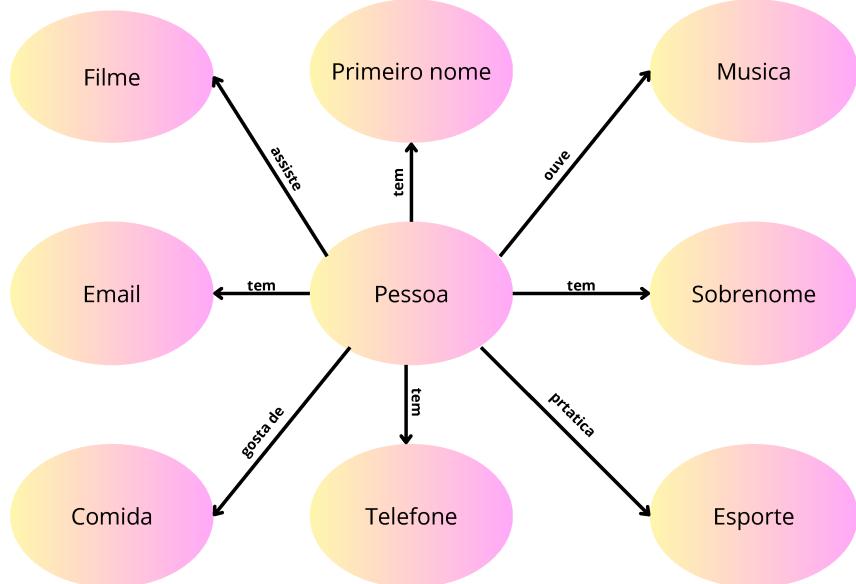
### 2.2.2 *Ontologias*

As ontologias exercem um papel central na Web Semântica ao fornecerem uma estrutura formal para organização do conhecimento e promoverem a interoperabilidade entre diferentes sistemas. As ontologias têm desempenhado um papel cada vez mais relevante na representação formal do conhecimento, promovendo a interoperabilidade entre sistemas e servindo de base para o desenvolvimento de aplicações inteligentes (BIBI *et al.*, 2023; STĂNESCU;

OPREA, 2025). Segundo Gruber (1993), uma ontologia é definida como “uma especificação explícita de uma conceitualização compartilhada”. Em termos práticos, isso significa oferecer um vocabulário comum para representar informações de forma estruturada e compreensível por máquinas, permitindo que diferentes agentes, sejam humanos ou computacionais, interpretem conceitos da mesma maneira (FENSEL *et al.*, 2020).

Para ilustrar de forma mais clara o que foi abordado até aqui, a Figura 4 apresenta uma representação gráfica de ontologia para humanos. Nesta representação, podemos observar como os conceitos “Pessoa” e “Música”, estão interligados por atributos e relacionamentos, formando um modelo compreensível e estruturado. No caso, uma “Pessoa” pode possuir atributos como “Primeiro Nome”, “Sobrenome” e “Email”, além disso, ela pode estabelecer relações com outros conceitos, como “Música”, indicando um gênero musical de sua preferência. Essa estrutura ajuda a organizar as informações de maneira clara e sem ambiguidades, formando a base para representações mais complexas.

Figura 4 – Representação Gráfica de ontologia para humanos



Fonte: Produzido pela autora.

Em uma ontologia, várias propriedades e componentes trabalham juntos para estruturar e organizar o conhecimento. Essas propriedades incluem (TAYE, 2010):

**Classes:** Representam categorias ou grupos de objetos que compartilham propriedades semelhantes. Elas funcionam como “tipos” de entidades em um domínio específico. Exemplo: A classe “Cidade” abrange todas as cidades, definindo propriedades gerais que qualquer instância de cidade deve ter.

**Instâncias:** As instâncias são os exemplos específicos de uma classe. Cada instância herda as propriedades da classe correspondente. Exemplo: “Sobral” e “Fortaleza” são instâncias da classe “Cidade”. Ambas herdam as propriedades da classe “Cidade”, como nome, estado, e população.

**Atributos:** Os atributos definem características de uma classe ou instância, podem ser considerados como as “colunas” de uma tabela em um banco de dados. Exemplo: Na classe “Cidade”, os atributos podem ser “nome”, “população” e “área”.

**Propriedades:** As propriedades estabelecem regras para esses atributos, determinando como eles se relacionam ou como devem se comportar dentro da ontologia. Exemplo: A propriedade “população” pode ter uma regra que determina que ela deve ser sempre um número inteiro maior que zero.

**Relacionamentos:** Os relacionamentos estruturam a ontologia ao conectar conceitos entre si, como “cidade pertence a um estado”. Exemplo: Um relacionamento entre as classes “Cidade” e “Estado” pode ser definido como “cidade pertence a um estado”. Isso significa que toda instância de “Cidade” estará vinculada a uma instância de “Estado”.

**Axiomas:** Os axiomas impõem restrições lógicas, garantindo a consistência da ontologia. Exemplo: Um axioma pode impor a regra de que “toda cidade deve pertencer a exatamente um estado”. Isso impede que instâncias de “Cidade” existam sem estar associadas a uma instância de “Estado” ou que estejam associadas a múltiplos estados.

Além de organizarem o conhecimento de forma clara, as ontologias têm grande importância em contextos computacionais. Elas são empregadas na modelagem de sistemas complexos, representando entidades e suas relações de maneira que possam ser processadas automaticamente. Um exemplo disso é o uso de triplas RDF (sujeito, predicado, objeto), que facilitam o intercâmbio semântico entre agentes. Tecnologias como OWL, serão discutidas na Seção 2.3.5, expandem essa lógica ao permitir a aplicação de regras e inferências mais sofisticadas (GUARINO *et al.*, 2009).

A construção de uma ontologia requer um vocabulário bem definido, estruturado logicamente para representar uma realidade específica de forma clara e compartilhável (DIAS; SANTOS, 2003). Para isso, duas propriedades fundamentais são destacadas:

- **Compartilhamento:** garante que diferentes aplicações e sistemas utilizem o mesmo vocabulário semântico, promovendo consistência na comunicação.
- **Filtragem:** permite selecionar apenas as informações relevantes para um determinado

contexto, abstraindo detalhes não essenciais.

Essas características tornam as ontologias poderosas aliadas na busca semântica. Diferentemente dos mecanismos tradicionais, como o Google, que operam por palavras-chave, os sistemas baseados em ontologias interpretam o significado dos termos e suas relações. Por exemplo, ao pesquisar “curso de programação avançado”, um buscador semântico entende que “programação” pode envolver linguagens como Python, Java ou JavaScript, e que “avançado” indica um nível de dificuldade. Essa compreensão é possível porque o sistema consulta uma ontologia que estrutura esses conceitos e suas inter-relações (TAYE, 2010).

Figura 5 – Tabela de Ontologia para "Curso de Programação Avançado"

| Classe                        | Subclasse                     | Relação                  | Instância                     |
|-------------------------------|-------------------------------|--------------------------|-------------------------------|
| Curso de Programação          | Curso de Programação Avançado | é do tipo de linguagem   | Python, JavaScript            |
| Curso de Programação          | Curso de Programação Avançado | tem nível de dificuldade | Avançado                      |
| Curso de Programação Avançado | -                             | é oferecido por          | Universidade Federal do Ceará |
| Curso de Programação Avançado | -                             | tem pré-requisito        | Conhecimento em algoritmo     |

Fonte: Produzido pela autora.

A ontologia apresentada na Figura 5 mostra como os conceitos de curso, linguagem, nível e modalidade estão interligados e organizados hierarquicamente, com subclasses e restrições associadas. Isso permite que os mecanismos semânticos filtrem cursos de acordo com critérios específicos e recomendem os mais adequados ao perfil do usuário.

Para exemplificar a aplicação prática de uma ontologia na descrição de um curso, é possível utilizar a ontologia *Schema.org*, amplamente adotada por mecanismos de busca e plataformas de conteúdo. O Listing 2.2.2 apresenta uma estrutura em *JSON-LD* sobre “Curso de Programação Avançado”, incorporando os elementos da Figura 5. Nesse modelo, a classe principal “Course” define o curso, enquanto suas características são representadas por propriedades como “educationalLevel” (nível de dificuldade), “coursePrerequisites” (pré-requisitos) e “teaches” (linguagens ensinadas). A instituição ofertante é especificada como uma “EducationalOrganization”, estabelecendo uma relação semântica com o provedor da formação.

Essa forma de representação permite que agentes computacionais compreendam com maior precisão o conteúdo, relacionem com as intenções dos usuários e ofereçam resultados mais relevantes em buscas ou sistemas de recomendação. O uso de ontologias como a do

*Schema.org* torna o conteúdo educacional mais interoperável, estruturado e acessível para diferentes plataformas na Web.

```

1  {
2      "@context": "https://schema.org",
3      "@type": "Course",
4      "name": "Curso de Programação Avançado",
5      "description": "Curso voltado ao aprofundamento em linguagens de programação
6      ↳ como Python e JavaScript, com foco em algoritmos e estruturas avançadas.",
7      "provider": {
8          "@type": "EducationalOrganization",
9          "name": "Universidade Federal do Ceará",
10         "url": "https://www.ufc.br"
11     },
12     "educationalCredentialAwarded": "Certificado de Conclusão",
13     "coursePrerequisites": "Conhecimento em algoritmo",
14     "hasCourseInstance": {
15         "@type": "CourseInstance",
16         "courseMode": "online",
17         "instructor": {
18             "@type": "Person",
19             "name": "Prof. João Pedro"
20         },
21         "startDate": "2025-08-01",
22         "endDate": "2025-12-15"
23     },
24     "audience": {
25         "@type": "EducationalAudience",
26         "educationalLevel": "Avançado"
27     },
28     "teaches": ["Python", "JavaScript"]
}

```

Listing 2.2.2 – Exemplo em JSON-LD para o Curso de Programação Avançado

A eficácia da Web Semântica será alcançada à medida que forem desenvolvidos programas capazes de realizar buscas precisas a partir de diversas fontes disponíveis na Web, correlacionando conteúdos com estruturas de conhecimento e regras de inferência. Com o aumento da disponibilidade de conteúdo estruturado, os agentes computacionais se tornam mais eficientes, sendo capazes de compreender as intenções dos usuários e oferecer resultados mais alinhados às suas necessidades (DIAS; SANTOS, 2003; HITZLER, 2021; HENDLER, 2001). Assim, diferentes propostas de classificação de ontologias na literatura reforçam o objetivo comum de criar vocabulários compartilhados, que viabilizem a troca de informações de forma clara e sem ambiguidade.

### 2.2.3 Agentes Inteligentes

Para que a comunicação entre a Web e o usuário ocorra de maneira eficiente, é necessário um modelo que permita a troca de informações sobre uma determinada tarefa. Nesse contexto, surgem os agentes inteligentes (ou agentes de software), programas que coletam, processam e compartilham informações para auxiliar os usuários. Em vez de executar todas as ações diretamente, esses agentes empregam técnicas de inteligência artificial visando auxiliar o usuário na realização de uma tarefa, oferecendo opções e sugestões, ajudando na tomada de decisões e tornando a interação com a Web mais eficiente (SOUZA; ALVARENGA, 2004).

Hendler (1999), em seu artigo, apresenta uma analogia com agentes de viagem para ilustrar o funcionamento dos agentes inteligentes da Web. Assim como um agente de viagens não realiza toda a tarefa por conta própria, mas auxilia na busca por opções, os agentes inteligentes interpretam os pedidos do usuário (como destino, datas e preferências), realizam buscas semânticas utilizando ontologias como FOAF (*Friend of a Friend*)<sup>9</sup> e Schema.org, consultam bases como *DBpedia* e *Wikidata*, e organizam os resultados de forma estruturada. A partir disso, sugerem rotas, meios de transporte, hospedagens e alternativas compatíveis com o perfil e orçamento do usuário.

Esses agentes empregam metadados para descrever e organizar as informações coletadas de diferentes fontes. Por exemplo, ao consultar uma base de dados de companhias aéreas, o agente acessa dados como preço, data, horário e política de bagagens, utilizando esses metadados para oferecer recomendações personalizadas. Além disso, agentes semânticos operam com base em regras e raciocínios complexos, utilizando ontologias e lógicas formais para interpretar o contexto e tomar decisões fundamentadas. Eles não apenas armazenam dados, mas compreendem as relações entre eles, possibilitando inferências que tornam as respostas mais relevantes e ajustadas às necessidades do usuário. Com isso, os agentes conseguem realizar associações sofisticadas, como:

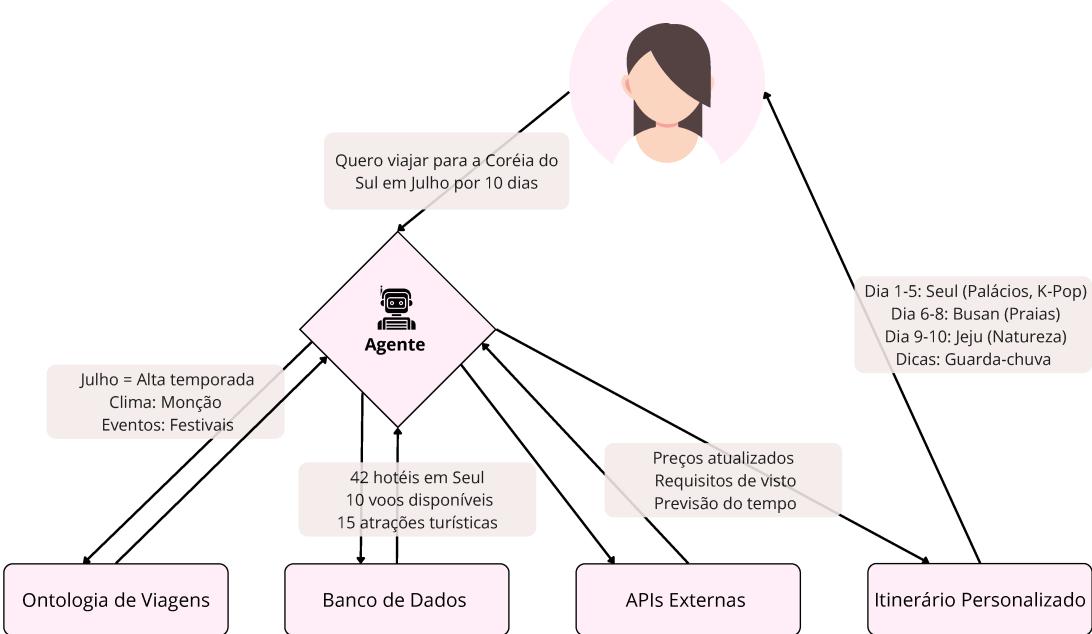
- Identificar que um hotel está localizado próximo ao local de realização de um evento.
- Reconhecer que atividades culturais específicas estão alinhadas ao perfil do viajante.
- Entender que uma viagem em julho para a Coreia exige atenção a aspectos climáticos e eventos sazonais.

A Figura 6 ilustra um exemplo de como essas conexões podem ser visualizadas em um grafo semântico:

---

<sup>9</sup> <<http://xmlns.com/foaf/spec/>>

Figura 6 – Representação de agentes inteligentes com base em grafos de conhecimento.



Fonte: Produzido pela autora.

Essa abordagem permite que agentes inteligentes não apenas recuperem dados relevantes, mas também organizem e apresentem essas informações de maneira contextualizada. Como resultado, as recomendações são mais precisas e personalizadas. A Web, nesse contexto, deixa de ser apenas um repositório passivo de páginas para se tornar um ambiente dinâmico e proativo, no qual sistemas computacionais compreendem as intenções dos usuários e agem de forma colaborativa.

Em síntese, os agentes inteligentes têm como função transformar a navegação em uma experiência mais intuitiva, servindo como consultores digitais. Eles utilizam inferência semântica, dados interconectados e regras lógicas para sugerir alternativas que atendam às necessidades específicas de cada usuário, tornando a tomada de decisão mais informada e eficaz (HENDLER, 1999; SOUZA; ALVARENGA, 2004).

#### 2.2.4 Web Services

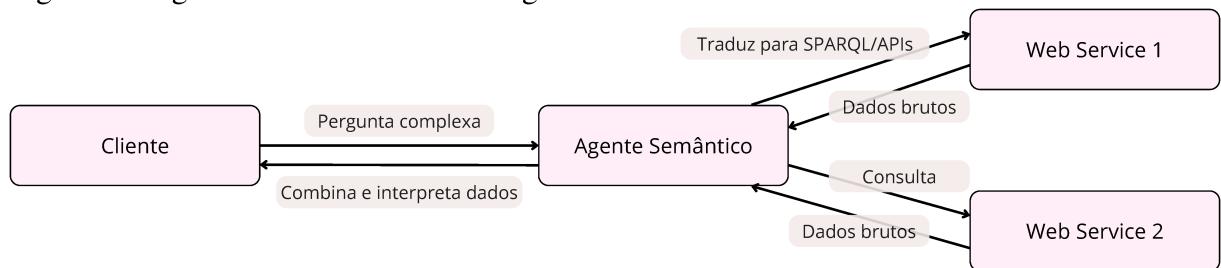
Web Services são os responsáveis por permitir a comunicação entre diferentes sistemas e a automação de processos. Eles são aplicações acessíveis pela internet que permitem a comunicação e troca de dados entre diferentes aplicativos ou plataformas pela internet, usando protocolos e padrões abertos, como *Hypertext Transfer Protocol* (HTTP), *Extensible Markup*

Language (XML), SOAP (Simple Object Access Protocol)<sup>10</sup> e REST (Representational State Transfer).

Na Web tradicional, os serviços funcionam com base em sintaxe, ou seja, exigem que os consumidores conheçam previamente as estruturas esperadas (parâmetros, formatos de retorno etc.). Já na Web Semântica, essa interação evolui para um nível mais sofisticado, pois os serviços podem ser descritos por meio de anotações semânticas que fornecem significado aos dados. Isso permite que agentes inteligentes descubram, interpretem, comparem e utilizem serviços de forma autônoma, sem a necessidade de codificação específica para cada caso (HENDLER, 2001).

A Figura 7 ilustra esse cenário, no qual o agente atua como uma camada de inteligência entre o cliente e diferentes serviços disponíveis. Ele interpreta uma solicitação complexa feita pelo usuário, consulta diversas fontes, traduz formatos, agrupa informações brutais e retorna uma resposta estruturada e personalizada. Essa funcionalidade só é possível devido ao uso de ontologias, metadados semânticos e raciocínio automatizado.

Figura 7 – Agente como “Camada Inteligente” sobre web services



Fonte: Produzido pela autora.

Para representar semanticamente os serviços, são utilizados padrões como OWL-S (Web Ontology Language for Services)<sup>11</sup>, que fornece uma ontologia em OWL para descrever serviços web em três aspectos: perfil (o que o serviço faz), modelo de processo (como ele funciona) e ligação (como acessá-lo). Outra abordagem comum é o uso de SAWSDL (Semantic Annotations for WSDL and XML Schema)<sup>12</sup>, que permite associar conceitos ontológicos diretamente a descrições WSDL existentes. Há ainda iniciativas como o WSMO (Web Service Modeling Ontology), voltadas à modelagem semântica, mediação e execução de serviços.

Essas tecnologias tornam possível a criação de sistemas com descoberta automática de serviços baseada em significado. Por exemplo, um agente pode procurar serviços que

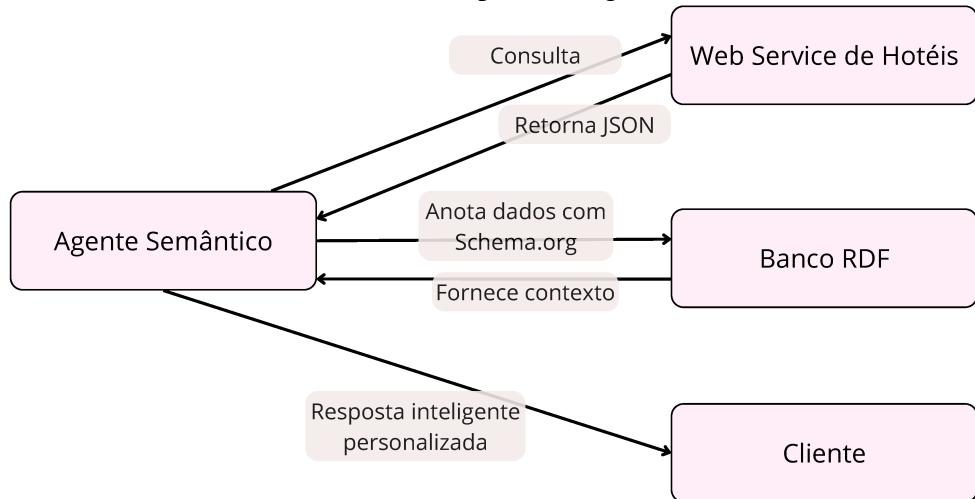
<sup>10</sup> <<https://www.w3.org/TR/soap/>>

<sup>11</sup> <<https://www.w3.org/submissions/OWL-S/>>

<sup>12</sup> <<https://www.w3.org/TR/sawsdl/>>

ofereçam “hospedagem próxima a pontos turísticos” mesmo que o termo usado pelo fornecedor seja “alojamento perto de atrações locais”. A ontologia e os metadados semânticos resolvem essa diferença terminológica.

Figura 8 – Web Service como fonte de dados para um agente



Fonte: Produzido pela autora.

A Figura 8 mostra um exemplo de aplicação prática: um agente semântico consulta um Web Service de hotéis que retorna dados em JSON. O agente anota semanticamente essas informações com vocabulários como Schema.org, armazena em um banco RDF e combina com outras fontes para responder ao cliente de forma personalizada. Essa capacidade é essencial em contextos como turismo, saúde, logística e comércio eletrônico, onde múltiplas fontes e critérios precisam ser integrados dinamicamente.

Para tornar essas informações comprehensíveis por máquinas, os serviços podem ser enriquecidos com metadados semânticos por meio de anotações como o vocabulário do Schema.org. Por exemplo, o Listing 2.2.3 apresenta uma possível resposta anotada em JSON-LD por um Web Service de hotéis.

```

1  {
2      "@context": "https://schema.org",
3      "@type": "Hotel",
4      "name": "Hotel Seul Garden",
5      "address": {
6          "@type": "PostalAddress",
7          "addressLocality": "Seul",
8          "addressCountry": "KR"
9      },
10     "starRating": {
11         "@type": "Rating",
12         "ratingValue": "4"
13     },
14     "amenityFeature": [
15         {
16             "@type": "LocationFeatureSpecification",
17             "name": "Wi-Fi",
18             "value": true
19         },
20         {
21             "@type": "LocationFeatureSpecification",
22             "name": "Café da manhã incluso",
23             "value": true
24         }
25     ]
26 }

```

Listing 2.2.3 – Exemplo em JSON-LD de um Web Service no domínio de hotéis

Com essa anotação, um agente semântico pode identificar automaticamente que se trata de um hotel com 4 estrelas em Seul, que oferece Wi-Fi e café da manhã, mesmo que essas informações estejam expressas de formas diferentes na interface do serviço. Isso permite que o agente combine esses dados com preferências do usuário e outras fontes, oferecendo respostas mais relevantes e personalizadas.

Além disso, a utilização de ontologias nas descrições dos serviços viabiliza o *matchmaking* semântico, ou seja, a correspondência inteligente entre as necessidades do usuário e as funcionalidades disponíveis. A Figura 9 exemplifica esse fluxo, destacando os papéis dos solicitantes, provedores e mecanismos de correspondência baseados em ontologias. Nessa arquitetura, tanto o consumidor quanto o fornecedor do serviço descrevem seus interesses ou capacidades por meio de anotações compatíveis com uma ontologia comum. O mecanismo de correspondência interpreta semanticamente essas descrições e identifica serviços relevantes

mesmo com vocabulários distintos.

Figura 9 – Ontologia em serviço na Internet



Fonte: Produzido pela autora.

Portanto, os web services semânticos representam um avanço significativo em relação aos modelos tradicionais. Eles permitem descoberta, composição e invocação automática de serviços com base em significados compartilhados. Quando combinados com agentes inteligentes, possibilitam a criação de sistemas flexíveis, autônomos e interoperáveis, capazes de transformar a Web em uma verdadeira plataforma de cooperação entre máquinas (HENDLER, 2001).

### 2.2.5 *Linked Data*

Linked Data, ou Dados Ligados, é uma abordagem que utiliza a Web para estabelecer conexões estruturadas entre diferentes conjuntos de dados. Essas informações podem vir de diversas fontes, como bancos de dados de organizações situadas em locais distintos ou até mesmo de sistemas internos dentro de uma mesma empresa, que antes não se comunicavam facilmente. A vasta quantidade de informações não estruturadas disponíveis na Internet impulsionou o desenvolvimento e a adoção de técnicas da Web Semântica, com o objetivo de tornar os dados mais interconectados, compreensíveis e úteis ao domínio público (AGUILAR *et al.*, 2023).

De forma técnica, Linked Data se refere a dados publicados na Web de maneira que possam ser processados automaticamente por máquinas. Esses dados possuem significados bem definidos, são interligados a outros conjuntos de informações externas e podem ser acessados e conectados por novos conjuntos de dados, criando uma grande rede de conhecimento interconectado (BIZER *et al.*, 2023).

Ele se baseia na utilização de URIs, que são necessários para identificar e vincular exclusivamente recursos na web. Um URI é basicamente uma *string* que identifica uma coisa ou recurso na Web. Todos os recursos de informação, como páginas WWW e documentos, possuem um *Uniform Resource Identifier* (URI) que é comumente conhecido como *Uniform Resource Locator* (URL). Isso facilita a navegação e a descoberta de informações interligadas em diferentes fontes, promovendo a criação de uma rede de dados semânticos.

Conforme discutido no artigo *Linked Data - The Story So Far*, de Bizer *et al.* (2008), essa abordagem é fundamental para a construção de uma Web Semântica, onde dados podem ser facilmente acessados e integrados por meio de diferentes plataformas e contextos. Um dos conjuntos de dados mais conhecidos dessa abordagem é o *DBpedia*, um projeto que extrai dados estruturados da Wikipédia e os disponibiliza como Linked Data. Isso permite que diferentes aplicações consultem e relacionem informações automaticamente. Por exemplo, ao consultar a DBpedia sobre um filme, é possível obter automaticamente dados sobre o elenco, diretor e ano de lançamento, sem depender de um sistema fechado.

Com o objetivo de estruturar essa publicação de dados de forma eficiente, Berners-Lee *et al.* (2006) estabeleceu um conjunto de princípios fundamentais para a disseminação de Linked Data. Esses princípios garantem que todos os dados publicados façam parte de um espaço de dados global único (HEATH, 2011):

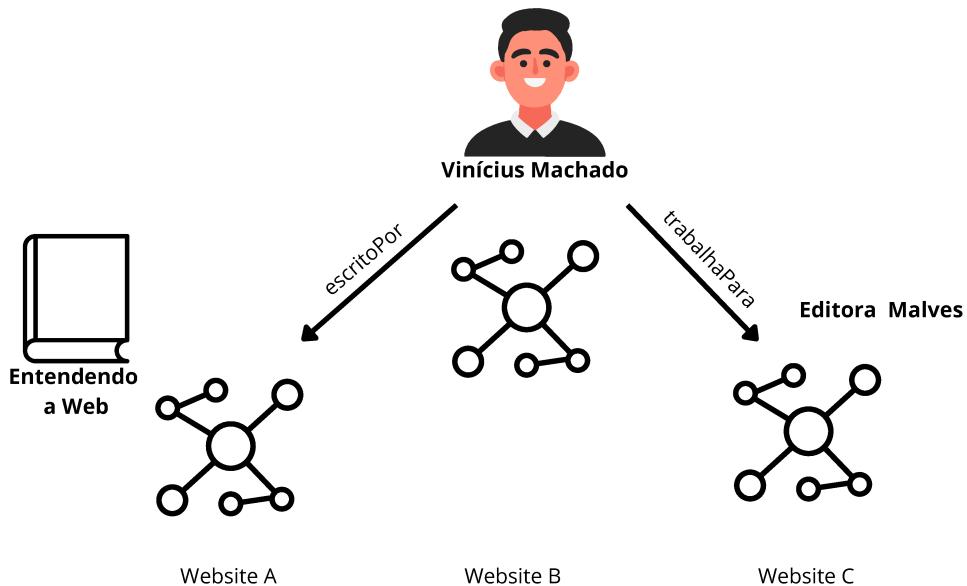
- Utilizar URIs para identificar recursos: cada entidade do mundo real (como pessoas, lugares ou conceitos) deve possuir um identificador único na Web.
- Utilizar URIs acessíveis via HTTP: essas URIs devem ser recuperáveis por meio do protocolo HTTP.
- Fornecer dados úteis ao acessar uma URI: o servidor deve retornar informações estruturadas em formatos padronizados, como RDF ou JSON-LD.
- Incluir links para outros dados: os dados disponibilizados devem conter referências para outros recursos, permitindo que humanos e agentes computacionais descubram novas informações de forma automática.

Essas diretrizes, conhecidas como os princípios do Linked Data, servem como um guia básico para a publicação e conexão de dados na Web. Ao respeitar a arquitetura e os padrões da Web, esses princípios possibilitam a criação de um ecossistema de dados interligados, promovendo maior acessibilidade e reutilização das informações.

A Figura 10 apresenta como exemplo um URI de identificador de recurso para um

livro chamado *Entendendo a Web*. A URI que identifica esse livro representa o recurso de forma única na Web. Quando acessada por um navegador habilitado para RDF, essa URI redireciona para o documento RDF que contém dados estruturados sobre o livro, incluindo propriedades como autor, data de publicação e relacionamentos com outras obras. Por outro lado, ao ser acessada por um navegador da Web HTML, a URI redireciona para um documento HTML, como uma página da Web sobre o livro no site da editora.

Figura 10 – Funcionamento do LinkedData



Fonte: Adaptado de (Wordlift, 2025).

O Linked Data se destaca como um dos fundamentos principais para a organização e recuperação estruturada de informações na Web Semântica. Sua aplicação viabiliza a interligação de fontes diversas, permitindo consultas mais precisas e contextualizadas. Diferente de bancos de dados isolados, essa abordagem facilita a descoberta de novos relacionamentos entre os dados, impulsionando a criação de serviços mais inteligentes e eficientes (HEATH, 2011). Dessa forma, o Linked Data não apenas melhora a acessibilidade da informação, mas também contribui para um ambiente digital mais integrado e dinâmico.

### 2.3 Ferramentas e Tecnologias da Web Semântica

A Web Semântica é fundamentada em uma série de tecnologias e padrões que possibilitam a descrição, a interconexão e o raciocínio sobre dados na Web. Essas tecnologias tornam possível a transformação de dados brutos, estruturados ou não, em conhecimento que pode ser compreendido e processado por máquinas, permitindo maior interoperabilidade entre

sistemas e agentes inteligentes.

Essas ferramentas podem ser organizadas em categorias complementares, conforme descrito a seguir:

**1. Modelos de dados:**

- Resource Description Framework (RDF)
- JSON-LD (formato JSON para RDF)
- XML (para serialização, como em RDF/XML)

**2. Linguagens de ontologia:**

- RDF Schema (RDFS)
- Web Ontology Language (OWL)
- Simple Knowledge Organization System (SKOS)

**3. Validação:**

- Shapes Constraint Language (SHACL)

**4. Consulta:**

- Protocol and Resource Description Framework Query Language (SPARQL)

### **2.3.1 *Resource Description Framework (RDF)***

O Resource Description Framework é um modelo criado pelo *World Wide Web Consortium* (W3C) para organizar e descrever informações na web de um jeito que tanto pessoas quanto máquinas possam entender. Ele é uma parte essencial da Web Semântica, pois ajuda a estruturar e dar significado aos dados, tornando mais fácil para os computadores processarem e relacionarem as informações. A ideia principal do RDF é representar os dados como triplas, em vez de armazenar dados de forma tradicional (como em tabelas de banco de dados), utiliza um formato baseado em relações, onde cada informação é conectada a outra. Para isso, ele usa um modelo simples de sujeito, predicado e objeto que liga elementos entre si, formando redes de conhecimento (DZIEKANIAK; KIRINUS, 2004).

Utilizando os mesmos dados apresentados na Figura 4 de representação de ontologia para humanos, segue a representação de uma estrutura formal destinada a computadores, onde as informações são expressas por meio de triplas <sujeito, predicado, objeto>. Essa estrutura de triplas, utilizada pelo RDF, permite a modelagem de conceitos e seus relacionamentos, alinhando-se ao modelo de ontologias da Web Semântica (MANOLA *et al.*, 2004; DZIEKANIAK; KIRINUS, 2004). A Tabela 3 descreve os três componentes fundamentais

que compõem uma tríplice RDF. Cada tríplice representa uma unidade de conhecimento, onde o sujeito identifica um recurso, o predicado expressa uma propriedade ou relação, e o objeto representa o valor ou recurso associado.

Tabela 3 – Componentes de uma Tríplice RDF

| Triplas   | Definição                                                                                                                                                                                                                                                                | Exemplo    |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| Sujeito   | “Pessoa”, ou algum objeto, ideia, ou seja, qualquer coisa que possa ser reconhecida, nomeada ou encontrada.                                                                                                                                                              | “Pessoa”   |
| Predicado | O predicado descreve a relação entre o sujeito e o objeto. Ele especifica como o sujeito está relacionado ao objeto. Os predicados são representados por <i>Uniform Resource Identifiers</i> (URIs) que apontam para termos definidos em vocabulários ou ontologias RDF. | "gosta de" |
| Objeto    | O objeto é o valor que está sendo relacionado ao sujeito, podendo ser um valor direto como um número, uma palavra ou uma referência para outro recurso.                                                                                                                  | "Comida"   |

Fonte: Produzido pela autora.

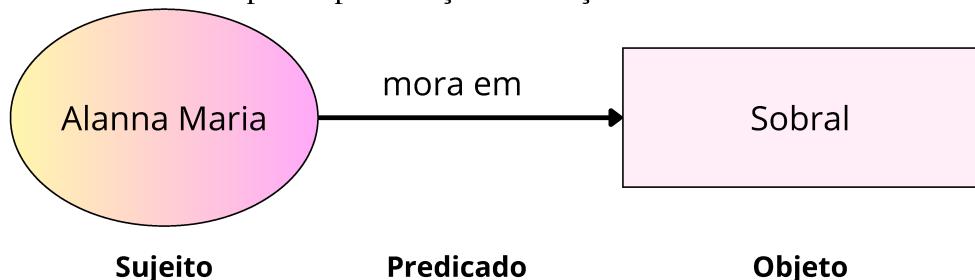
Essa estrutura básica de sujeito, predicado e objeto pode ser usada para descrever relações simples e diretas, como “Pessoa → gosta de → Comida”.

Para ilustrar um exemplo prático mais complexo sobre uma “Pessoa” com várias informações associadas, considere o seguinte caso:

- <“Pessoa”, “tem nome”, “Alanna Maria”>
- <“Pessoa”, “trabalha em”, “Empresa XYZ”>
- <“Pessoa”, “mora em”, “Sobral”>

Essas triplas formam um grafo RDF que descreve a pessoa “Alanna Maria”, permitindo que sistemas computacionais compreendam e processem. O sujeito “Pessoa” está relacionado ao objeto de “Sobral” por meio da ação “mora em”, formando uma relação clara e estruturada. Para representar graficamente, podemos utilizar um grafo de relações entre conceitos interligados (MANOLA *et al.*, 2004; GUTIERREZ *et al.*, 2006).

Figura 11 – Estrutura RDF para representação de relações entre conceitos



Fonte: Produzido pela autora.

Na Figura 11, os recursos são representados por elipses e os valores literais por retângulos. O RDF adota uma semântica bem definida, na qual cada tripla constitui um grafo orientado, com o sujeito como nó de origem, o objeto como nó de destino, e o predicado como a aresta nomeada entre eles.

Além da representação gráfica, o RDF pode ser serializado em diferentes formatos, sendo o RDF/XML o primeiro padrão definido. Esse formato facilita a interoperabilidade entre sistemas ao padronizar a descrição de dados. Aplicações típicas incluem redes sociais semânticas, integração de contatos em sistemas distribuídos e processamento automático de informações pessoais (SOUZA; ALVARENGA, 2004).

Outra alternativa para representá-lo são os formatos de serialização, sendo o RDF/XML o primeiro padrão estabelecido. Desenvolvido como sintaxe original para RDF, esse formato facilita a interoperabilidade entre diferentes sistemas ao descrever dados de maneira padronizada e sem ambiguidades. Aplicações comuns incluem redes sociais semânticas, integração de dados de contato em sistemas distribuídos e aplicações que precisam processar contatos de pessoas automaticamente (SOUZA; ALVARENGA, 2004). O Listing 2.3.1 apresenta um exemplo de um documento com anotações semânticas:

```

1  <?xml version="1.0"?>
2  <rdf:RDF
3      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4      xmlns:foaf="http://xmlns.com/foaf/0.1/">
5
6      <foaf:Person rdf:about="#alanna">
7          <foaf:name>Alanna Maria</foaf:name>
8          <foaf:mbox rdf:resource="mailto:alanna.maría@gmail.com"/>
9          <foaf:age>25</foaf:age>
10         <foaf:gender>female</foaf:gender>
11     </foaf:Person>
12 </rdf:RDF>

```

Listing 2.3.1 – Exemplo de serialização RDF/XML usando vocabulário FOAF

Esse código apresenta:

**Linha 1:** Declaração XML padrão.

**Linhas 2-3:** Definição dos namespaces:

- **rdf:** - Namespace básico do RDF.
- **foaf:** - Vocabulário FOAF para descrição de pessoas.

**Linha 4:** Instância de Pessoa **foaf:Person** com:

- **rdf:about** - URI único identificando o recurso.

**Linha 5:** Propriedade **foaf:name** com valor literal, nome completo.

**Linha 6:** Propriedade **foaf:mbox** vinculada a recurso externo via URI.

**Linha 7:** Propriedade **foaf:age** com valor literal, idade.

**Linha 8:** Propriedade **foaf:gender** com valor padronizado, gênero.

**Linha 9:** Fechamento da estrutura RDF.

Apesar de sua utilidade, o RDF/XML apresenta desafios como verbosidade e menor legibilidade em comparação com outras formas de serialização, como Turtle.

A evolução dessa estrutura levou ao conceito de Linked Data, que foca na interligação de conjuntos de dados por meio de URIs. Essa abordagem permite conexões dinâmicas entre informações diversas, cada uma identificada de forma única na Web. O projeto DBpedia, extraído da Wikipédia, é um dos mais conhecidos e serve como ponto central da Linked Open Data Cloud, com milhões de entidades e bilhões de triplas interconectadas (HITZLER, 2021).

O RDF é especialmente útil nesse contexto por sua capacidade de descrever qualquer entidade, desde que possua um identificador na Web (DZIEKANIAK; KIRINUS, 2004; SILVA, 2015). Por exemplo:

- <<https://exemplo.com/recursos/Alanna>>, representa “Alanna”, a pessoa descrita.
- <<https://exemplo.com/propriedades/moraEm>>, define a propriedade “mora em” (propriedade).
- <<https://exemplo.com/recursos/Sobral>>, identifica a cidade “Sobral”, onde Alanna mora.

Assim, o RDF se estabelece como um modelo simples, porém poderoso, para a descrição semântica de dados na Web, permitindo integração, compartilhamento e interpretação automatizada de informações em larga escala.

### 2.3.2 *RDF Schema (RDFS)*

O RDFS (RDF Schema) é uma ampliação do RDF que permite a definição de vocabulários e a criação de estruturas de classes e propriedades em um modelo RDF. Ambas as tecnologias foram desenvolvidas para promover a criação e compartilhamento descentralizado de informações e vocabulários na web. Assim como a web possibilita que qualquer usuário crie uma página e a conecte a outras, o RDFS segue o mesmo princípio, permitindo que qualquer pessoa expresse informações sobre qualquer tema. Enquanto o RDF se concentra na descrição básica de recursos, o RDFS incorpora semântica e hierarquia, proporcionando uma organização mais detalhada e estruturada dos dados (GUTIERREZ *et al.*, 2006; MCBRIDE, 2004).

A Tabela 4 apresenta os principais elementos da estruturação do RDFS. Esses componentes permitem a criação de vocabulários mais expressivos, a definição de classes, propriedades e hierarquias, bem como a adição de rótulos e comentários que facilita o entendimento humano.

Tabela 4 – Recursos e suas características em RDF

| Elemento do RDFS      | Descrição                                                      | Exemplo(s)                                                                                                                                         |
|-----------------------|----------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| Classes               | Permite categorizar recursos em diferentes tipos.              | rdfs:Class define que um recurso é uma classe.                                                                                                     |
| Propriedades          | Definem relações entre os recursos.                            | rdfs:domain e rdfs:range especificam o tipo de recurso que pode ser usado como sujeito ou objeto de uma propriedade.                               |
| Herança               | Permite a definição de hierarquias de classes e propriedades.  | rdfs:subClassOf: define que uma classe é uma subclasse de outra.<br>rdfs:subPropertyOf: define que uma propriedade é uma sub propriedade de outra. |
| Rótulos e comentários | Proporciona descrições textuais e nomes legíveis para humanos. | rdfs:label: fornece um rótulo amigável para o recurso.<br>rdfs:comment: permite adicionar comentários sobre o recurso ou propriedade.              |
| Tipo de recurso       | Define o tipo de um recurso específico.                        | rdf:type indica que um recurso é uma instância de uma classe.                                                                                      |

Fonte: Produzido pela autora.

Com base nesses elementos, é possível estruturar semanticamente uma descrição para a pessoa “Alanna Maria”. Primeiro, define-se a classe “Pessoa” com o uso de rdfs:Class. Em seguida, cria-se a instância ex:Alanna como um recurso do tipo “Pessoa” utilizando rdf:type. As propriedades como “hasName”, “hasAgehasGender” são então associadas a essa instância, cada uma com seus domínios e intervalos definidos:

- A propriedade hasName tem como domínio a classe Pessoa e como intervalo um literal do tipo xsd:string;
- A propriedade hasAge tem como intervalo xsd:integer;
- A propriedade hasGender pode ter como intervalo uma instância de uma classe chamada, por exemplo, Gênero.

Assim, ao criar o recurso ex:Alanna, atribui-se:

- ex:Alanna rdf:type ex:Pessoa
- ex:Alanna ex:hasName “Alanna Maria”

- ex:Alanna ex:hasAge “25”@sd:integer
- ex:Alanna ex:hasGender “Feminino”

Esse modelo não apenas estrutura os dados de forma clara, mas também estabelece suas relações semânticas, tornando possível o compartilhamento e a reutilização dos dados em diferentes contextos e por diferentes aplicações (DECKER *et al.*, 2000).

### 2.3.3 *JavaScript Object Notation for Linked Data (JSON-LD)*

O JSON-LD (JavaScript Object Notation for Linked Data) é uma tecnologia voltada para a representação de dados estruturados semântica, seguindo os princípios do Linked Data, ou seja, é a implementação prática. Desenvolvido para facilitar a publicação e o consumo desses dados na web, o JSON-LD utiliza o formato simples e legível do JSON, que é amplamente usado em APIs e na Web oferecendo uma maneira simples e eficaz de descrever informações de forma semanticamente rica. Sua estrutura permite identificar recursos por meio de URIs e descrever relações entre eles usando o modelo RDF, o que garante a troca de informações entre diferentes aplicações e serviços, promovendo uma web mais conectada (CONSORTIUM, 2020).

Além disso, JSON-LD, é possível adicionar metadados enriquecidos a páginas da web (como informações de produtos, eventos e perfis), permitindo que mecanismos de busca e sistemas semânticos processem melhor os conteúdos (CONSORTIUM, 2020). Por exemplo, o JSON-LD permite que páginas da web como o Google informe melhor os mecanismos de busca sobre o conteúdo da página, ajudando a exibir informações como classificações, eventos, receitas ou perfis de empresas diretamente nos resultados de pesquisa. Portanto, ele não só está alinhado com os conceitos da Web Semântica, como também ajuda a torná-los mais acessíveis e práticos para desenvolvedores, um exemplo disso pode ser visualizado no Listing 2.3.2.

```

1 {
2   "@context": "https://schema.org",
3   "@type": "Person",
4   "name": "Guiomar Machado",
5   "knows": {
6     "@type": "Person",
7     "name": "Mairton Paiva"
8   }
9 }
```

Listing 2.3.2 – Exemplo prático de JSON-LD aplicando princípios de Linked Data

### 2.3.4 Extensible Markup Language (XML)

Segundo Consortium *et al.* (2006) no artigo “*Extensible Markup Language (XML) 1.1 (Second Edition)*”: A Linguagem de Marcação Extensível (XML), foi projetada principalmente para representar dados de forma hierárquica e estruturada. Essa linguagem descreve documentos, usados com frequência para armazenar e transportar dados. Além disso, oferece uma descrição parcial do comportamento dos softwares que processam esses dados. Um módulo de software, denominado processador XML, é responsável pela leitura de documentos e pelo fornecimento de acesso ao seu conteúdo e estrutura.

A estrutura hierárquica organiza os dados em elementos contidos em tags, sendo que cada uma possui um significado específico dentro da estrutura. Como o XML não incorpora semântica interna, a interpretação dos dados depende do contexto da aplicação. Embora seja amplamente utilizado na web, não está exclusivamente direcionado à Web Semântica (ALMEIDA, 2002). Um exemplo de fragmento XML é apresentado no Listing 2.3.3:

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <pessoa>
3      <nome>Alanna Maria</nome>
4      <email>alanna.maría@example.com</email>
5  </pessoa>

```

Listing 2.3.3 – Exemplo simplificado de um fragmento XML

### 2.3.5 Web Ontology Language (OWL)

A Web Ontology Language é uma linguagem essencial para a representação de ontologias na Web Semântica. Seu formato original, baseado na sintaxe RDF/XML, embora amplamente adotado, apresenta desafios significativos de legibilidade devido à complexidade dessa estrutura.

Essa limitação prática levou ao surgimento de alternativas sintáticas mais acessíveis. Uma delas é a sintaxe baseada em XML simplificada, que mantém a estrutura em tags, mas elimina as regras rígidas do RDF, tornando-se mais intuitiva e legível para humanos. Outra é a sintaxe abstrata, usada na documentação formal da linguagem, por ser mais compacta e adequada a contextos técnicos. Há ainda a sintaxe gráfica baseada em UML (Unified Modeling Language), que aproveita a familiaridade dos desenvolvedores com modelagem para facilitar a

compreensão de ontologias e a comunicação entre stakeholders. A existência dessas variações visa acomodar os construtos semânticos da OWL (ANTONIOU; HARMELEN, 2009). Seus componentes básicos incluem:

**Classes:** São estruturas hierárquicas que organizam conceitos por meio da propriedade `rdfs:subClassOf`. Exemplo: No Listing 2.3.4 mostra que Professor é subclasse de Pessoa → todo Professor é uma Pessoa.

```

1 <owl:Class rdf:about="http://exemplo#Pessoa">
2   <rdfs:label>Pessoa</rdfs:label>
3 </owl:Class>
4
5 <owl:Class rdf:about="http://exemplo#Professor">
6   <rdfs:subClassOf rdf:resource="http://exemplo#Pessoa"/>
7 </owl:Class>

```

Listing 2.3.4 – Exemplo de Classes

**Instâncias:** Indivíduos concretos que pertencem a classes. Exemplo: No Listing 2.3.5, Matheus é uma instância da classe Pessoa.

```

1 <owl:NamedIndividual rdf:about="http://exemplo#Matheus">
2   <rdf:type rdf:resource="#Pessoa"/>
3   <rdfs:label>Manoel Matheus</rdfs:label>
4 </owl:NamedIndividual>

```

Listing 2.3.5 – Instância de indivíduo em *Web Ontology Language* (OWL)

**Propriedades de Objeto:** Estabelecem relações entre indivíduos, podem ter características específicas, `rdfs:subPropertyOf`. Exemplo: No Listing 2.3.6, declara que temPai é uma propriedade de objeto (`owl:ObjectProperty`).

```

1 <owl:ObjectProperty rdf:about="http://exemplo#temMãe">
2   <rdfs:domain rdf:resource="http://exemplo#Pessoa"/>
3   <rdfs:range rdf:resource="http://exemplo#Pessoa"/>
4   <rdfs:label>tem mãe</rdfs:label>
5 </owl:ObjectProperty>

```

Listing 2.3.6 – Propriedade de objeto em OWL

**Propriedades de Dados:** Propriedades que vinculam indivíduos a valores literais. Exemplo: `xsd:string`, `xsd:date`, `xsd:integer`, com validação de tipos e intervalos, no

Listing 2.3.7 apresenta a demonstração de seu uso.

```

1 <owl:DatatypeProperty rdf:about="http://exemplo#temIdade">
2   <rdfs:domain rdf:resource="http://exemplo#Pessoa"/>
3   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#integer"/>
4   <rdfs:label>tem idade</rdfs:label>
5 </owl:DatatypeProperty>

```

Listing 2.3.7 – Propriedade de dados em OWL

Diferentemente de formatos básicos como XML ou mesmo RDF, a OWL foi desenvolvida para ser utilizada em contextos onde as informações de documentos precisam ser processadas por softwares, ao invés de apenas exibidas para seres humanos. É usada para criar e definir ontologias, ou seja, melhorar a capacidade das máquinas de entender o conteúdo da Web, oferecendo um vocabulário mais detalhado e uma semântica formal que facilita a interpretação automatizada dos dados (BECHHOFER *et al.*, 2004).

A OWL apresenta três sub-linguagens, cada uma sendo mais expressiva que a anterior: OWL Lite, OWL DL e OWL Full. Essas variantes expandem o vocabulário disponível para descrever propriedades e classes, incluindo relações entre classes, cardinalidade, igualdade, uma tipagem mais detalhada de propriedades, e características enumeradas para propriedades e classes. Além disso, o OWL oferece uma semântica formal que permite que máquinas realizem raciocínios sobre o significado do conteúdo, um recurso ausente em XML ou RDF. Esse aspecto facilita o processamento e a integração automatizados de informações disponíveis na Web, alinhando-se com o propósito central da Web Semântica (MCGUINNESS *et al.*, 2004; BECHHOFER *et al.*, 2004). A Tabela 5 contém uma breve descrição de cada um destes tipos de Sub-linguagem OWL:

Tabela 5 – Descrição das sub-linguagens do OWL

| Sub-linguagem   | Descrição                                                                                                                                                                                                                                                 |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>OWL Lite</b> | Sub-linguagem simples, projetada para aplicações que equilibram expressividade e complexidade, com suporte a hierarquia de classes, propriedades, restrições simples e tipos de dados básicos.                                                            |
| <b>OWL DL</b>   | Sub-linguagem intermediária, oferece maior expressividade, mantendo integridade computacional e decidibilidade.                                                                                                                                           |
| <b>OWL Full</b> | Sub-linguagem de alta complexidade, voltada para aplicações que exigem máxima expressividade e liberdade sintática do RDF, permitindo ontologias detalhadas sem preocupações com decidibilidade, usando lógica de primeira ordem e construções avançadas. |

Fonte: Produzido pela autora.

Como observado, essa arquitetura em camadas permite que a OWL atenda desde necessidades básicas até os requisitos mais exigentes de representação de conhecimento, sempre com o objetivo final de enriquecer a capacidade das máquinas de compreender e integrar informações na Web.

### 2.3.6 *Simple Knowledge Organization System (SKOS)*

O Simple Knowledge Organization System (SKOS) é um modelo de dados desenvolvido pela W3C com o objetivo de representar sistemas de organização do conhecimento, como tesouros, taxonomias, esquemas de classificação e vocabulários controlados de forma padronizada e interoperável na Web. Seu propósito principal é facilitar a representação, reutilização e o compartilhamento desses esquemas, especialmente no contexto da Web Semântica e do Linked Data. Surgiu como uma alternativa simplificada para adaptar os tesouros tradicionais. Enquanto os tesouros clássicos são amplamente utilizados na indexação e recuperação da informação, o SKOS oferece uma camada semântica especializada, baseada em RDF, que permite que esses sistemas sejam interpretados por máquinas e vinculados a outros dados na Web (MILES *et al.*, 2005).

Em outras palavras, o SKOS funciona como um vocabulário RDF específico para a organização do conhecimento, oferecendo estruturas prontas para uso. Já o RDF genérico exige que todos os elementos sejam definidos do zero. O modelo SKOS fornece um conjunto de classes e propriedades RDF que permitem representar conceitos e suas relações (como hierarquias e associações) em forma de grafos conectáveis na Web (MILES *et al.*, 2005), por exemplo:

- Conceitos (skos:Concept), como entidades básicas (ex.: Inteligência Artificial).
- Rótulos (skos:prefLabel para termos preferenciais, skos:altLabel para sinônimos).
- Relações semânticas (skos:broader/narrower para hierarquias, :related para associações).

Essas propriedades permitem modelar sistemas de organização do conhecimento tradicional na Web Semântica, como:

**Tesouros:** Utiliza skos:prefLabel para indicar o termo preferido de um conceito, skos:altLabel para representar sinônimos ou variantes, e skos:related para estabelecer relações associativas entre conceitos não hierárquicos, conforme observado no Listing 2.3.8.

```

1  :musica a skos:Concept ;
2    skos:prefLabel "Música"@pt ;
3    skos:altLabel "Som musical"@pt ;
4    skos:related :danca .

```

Listing 2.3.8 – Exemplo de tesauro em SKOS

**Taxonomias:** Estrutura conceitos em hierarquias com o uso de `skos:broader` para conceitos mais amplos e `skos:narrower` para conceitos mais específicos, conforme observado no Listing 2.3.9.

```

1  :animal a skos:Concept ;
2    skos:prefLabel "Animal"@pt ;
3    skos:narrower :mamifero .

```

Listing 2.3.9 – Exemplo de taxonomia em SKOS

**Glossários:** Aplica `skos:definition` para fornecer definições formais dos conceitos, conforme observado no Listing 2.3.10.

```

1  :internet a skos:Concept ;
2    skos:prefLabel "Internet"@pt ;
3    skos:definition "Rede global de computadores que permite a comunicação e
4      → troca de dados."@pt .

```

Listing 2.3.10 – Exemplo de glossário em SKOS

**Esquemas de Classificação:** Utiliza `skos:ConceptScheme` para organizar conjuntos de conceitos relacionados, e propriedades como `skos:hasTopConcept` e `skos:inScheme` para associá-los à estrutura geral, conforme observado no Listing 2.3.11.

```

1  web:taxonomiaWeb a skos:ConceptScheme ;
2    skos:prefLabel "Taxonomia de Tecnologias Web"@pt ;
3    skos:hasTopConcept web:frontend, web:backend, web:protocolos ;
4    skos:note "Classificação de tecnologias para desenvolvimento web
5      → moderno"@pt .
6
6  web:frontend a skos:Concept ;
7    skos:prefLabel "Frontend"@pt ;
8    skos:narrower web:html, web:css, web:javascript ;
9    skos:definition "Tecnologias executadas no navegador do usuário"@pt .

```

Listing 2.3.11 – Exemplo de esquema de classificação em SKOS

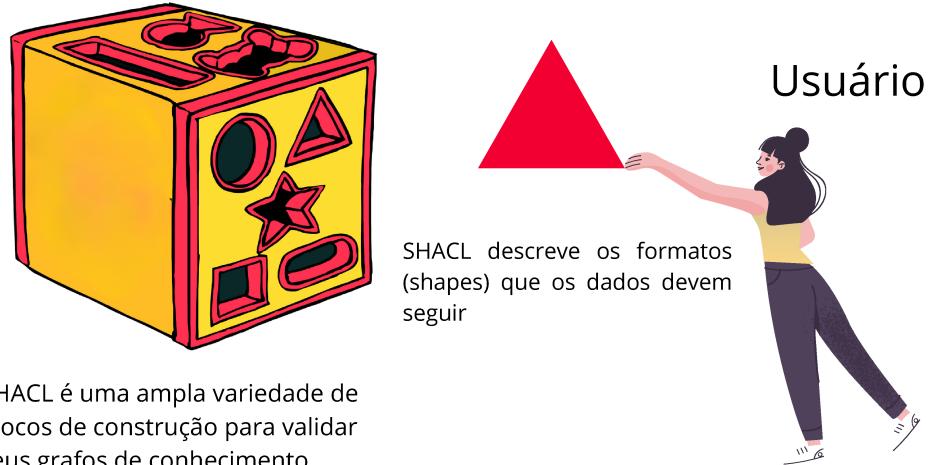
Além disso, disponibiliza uma linguagem simples e de fácil compreensão, adequada tanto para a criação quanto para o compartilhamento de novos esquemas de organização do conhecimento. Com o uso do SKOS, um sistema de organização do conhecimento pode ser representado como dados legíveis por máquinas. Isso permite sua troca entre diferentes aplicações e sua publicação na Web em um formato padronizado e processável automaticamente. O SKOS pode ser utilizado de forma independente ou em conjunto com linguagens mais formais de representação do conhecimento, como a OWL (MILES; BECHHOFER, 2009).

Embora tanto SKOS quanto OWL sejam utilizados para modelar conhecimento, seus propósitos e capacidade de representação são distintos. SKOS é otimizado para a representação de estruturas conceituais simples, como “Instrumento musical” é mais amplo que “Violino”, sem a necessidade de lógica formal. Por isso, é ideal para modelar vocabulários controlados e taxonomias com baixa complexidade semântica, sem ter a necessidade de criar propriedades personalizadas.

### 2.3.7 *Shapes Constraint Language (SHACL)*

O SHACL (Shapes Constraint Language) é uma linguagem padrão da W3C para descrever e validar gráficos RDF como ontologias, grafos de conhecimento ou *datasets* semânticos em relação a um conjunto de condições, surgiu com a necessidade de especificar restrições em grafos RDF e mecanismos para detectar violações dessas restrições. Essas condições são fornecidas como *shapes* (formas) e outros elementos representados como grafos RDF, esses grafos validados são chamados de *data graphs* (grafo de dados). O *Shapes Constraint Language* (SHACL) funciona como um “fiscal de qualidade”, assegurando que os dados em grafos RDF sigam regras específicas (validação da estrutura conforme o modelo esperado). Além disso, essas regras (*shapes*) também podem ser usadas para descrever grafos válidos, com aplicações que vão além da validação, como: Construção de interfaces de usuário, Geração de código e Integração de dados (W3C, 2017). Na Figura 12 é possível visualizar um exemplo análogo para compreender a definição.

Figura 12 – Analogia de representação do SHACL



Fonte: Adaptado de (Ontotext, 2025).

Ele é formado principalmente por 3 conceitos fundamentais (CORMAN *et al.*, 2018; W3C, 2017):

### Shapes e Targets

- **Shapes:** São definições que especificam as regras que os dados devem seguir, podem ser *Node Shapes* (aplicáveis a nós específicos, ex: sh:NodeShape) ou *Property Shapes* (aplicáveis a propriedades).
- **Targets:** Especificam onde as regras se aplicam, a quais nós do grafo RDF as *shapes* se aplicam (ex: sh:targetClass, aplica-se a todos os nós de uma classe, sh:targetNode, aplica-se a nós específicos, sh:targetSubjectsOf, aplica-se a sujeitos de uma propriedade.).

### Constraints (Restrições)

- **Restrições de propriedades**, como cardinalidade e tipo de dado, sh:minCount, sh:datatype, respectivamente.
- **Restrições lógicas**, como sh:and, sh:or e sh:not.
- **Restrições de valor**, como sh:hasValue e sh:in.
- **Restrições baseadas em SPARQL**, como sh:sparql.

### Validação e Relatórios

- Um validador SHACL verifica se os dados estão conforme as *shapes* definidas.
- Se uma violação for encontrada, um relatório detalhado é gerado, indicando onde e por que os dados não atendem às restrições.

Analizando o exemplo de código RDF no Listing 2.3.1 que descreve uma pessoa utilizando o vocabulário FOAF, onde é necessário garantir que todo indivíduo do tipo “foaf:Person”

possua exatamente uma propriedade “foaf:name” (obrigatória, do tipo xsd:string) e uma propriedade foaf:age opcional, que quando presente deve ser um valor inteiro (xsd:integer) e maior ou igual a zero (sh:minInclusive 0).

O Shape SHACL a seguir formaliza essas regras de validação:

```

1  @prefix sh: <http://www.w3.org/ns/shacl#> .
2  @prefix foaf: <http://xmlns.com/foaf/0.1/> .
3  @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

4
5  foaf:PersonShape
6      a sh:NodeShape ;
7      sh:targetClass foaf:Person ;

8
9      # Regra para o nome (obrigatório)
10     sh:property [
11         sh:path foaf:name ;
12         sh:minCount 1 ;
13         sh:maxCount 1 ;
14         sh:datatype xsd:string ;
15         sh:message "Erro: Cada pessoa deve ter exatamente um nome (string)." ;
16     ] ;
17
18      # Regra para idade (opcional, se existir)
19     sh:property [
20         sh:path foaf:age ;
21         sh:datatype xsd:integer ;
22         sh:minInclusive 0 ;
23         sh:message "Erro: Idade deve ser um inteiro positivo." ;
24     ] ;

```

Listing 2.3.12 – Shape SHACL para validação de dados pessoais em FOAF

O Listing 2.3.12 apresentado do *Shape SHACL* para validação de dados pessoais ilustra de maneira concreta o poder desta linguagem na garantia da qualidade de dados semânticos. Como demonstrado no código, a validação opera em três níveis fundamentais (CORMAN *et al.*, 2018; W3C, 2017):

**Validação Estrutural:** Através das propriedades sh:minCount e sh:maxCount, o *shape* impõe restrições cardinais, garantindo que todo indivíduo do tipo foaf:Person possua exatamente um nome. Essa validação previne problemas comuns como dados incompletos ou duplicados.

**Validação de Tipagem:** O uso explícito de sh:datatype com tipos XML Schema (xsd:string, xsd:integer) assegura a integridade sintática dos valores, evita inconsistências como

números representados como strings ou formatos de data inválidos.

**Validação Semântica:** A restrição sh:minInclusive aplicada à propriedade foaf:age demonstra como SHACL vai além da sintaxe ao incorporar regras de negócio diretamente no processo de validação.

Quando aplicado a grafos RDF, este mecanismo gera relatórios de validação detalhados que cumprem um papel duplo: técnico e documental. Cada violação é acompanhada de mensagens claras (como “Erro: Idade deve ser um inteiro positivo”), que servem tanto para correção imediata quanto para documentação viva das regras de qualidade. O SHACL é uma ferramenta poderosa para assegurar a qualidade de dados semânticos, sendo amplamente adotado em projetos de dados vinculados e ontologias. Seu uso facilita a detecção de inconsistências e a manutenção de conjuntos de dados confiáveis.

### 2.3.8 *Protocol and Resource Description Framework Query Language (SPARQL)*

SPARQL Protocol and RDF Query Language é uma linguagem de consulta padrão definida pela W3C para consultar bases de dados que armazenam dados no formato RDF. Com SPARQL, é possível realizar consultas complexas, como recuperar dados, fazer junções entre diferentes conjuntos de dados e filtrar informações, usando uma sintaxe semelhante a SQL, mas adaptada ao modelo de grafos do RDF. Ele pode ser usado para consultar fontes de dados como triplas RDF e endpoints de dados na web, como ontologias, dados de linked data e bases de dados de conhecimento (PÉREZ *et al.*, 2009).

Uma consulta típica em SPARQL compreende padrões de triplas, conectivos lógicos, opções e padrões opcionais. Atualmente, existem quatro tipos distintos de consultas (SILVA, 2015):

- **SELECT Query:** Usada para recuperar informações diretamente de um ponto de acesso SPARQL. Os resultados são apresentados em formato tabular, no estado em que os dados estão armazenados.
- **CONSTRUCT Query:** Permite extrair informações de um ponto de acesso e transformá-las em um formato RDF válido, reestruturando os dados para fins específicos.
- **ASK Query:** Retorna um valor booleano (verdadeiro ou falso) com base em uma consulta executada em um ponto de acesso SPARQL, indicando se a condição especificada é verdadeira ou não.
- **DESCRIBE Query:** Usada para obter um exemplo de grafo RDF de um ponto de acesso

SPARQL. O conteúdo exato do grafo depende do ponto de acesso consultado, podendo variar conforme a implementação.

O Listing 2.3.13 apresenta a representação de uma consulta SPARQL do tipo *SELECT* que retorna os nomes e endereços de e-mail de pessoas no grafo RDF que usam o namespace *FOAF* (*Friend of a Friend*).

```

1 PREFIX foaf: <http://xmlns.com/foaf/0.1/>
2   SELECT ?name ?email
3   WHERE {
4     ?person foaf:name ?name ;
5       foaf:mbox ?email .
6 }
```

Listing 2.3.13 – Exemplo de uma consulta SPARQL do tipo SELECT

A cláusula “PREFIX” define um prefixo para simplificar as consultas ao namespace FOAF. A cláusula “SELECT” especifica as variáveis a serem recuperadas na consulta. A cláusula “WHERE” define os padrões que os dados devem corresponder para serem recuperados, onde “?person” tem um nome e um endereço de e-mail associados a ele no grafo RDF.

## 2.4 Web Semântica no cenário atual

A Web Semântica continua a ser uma área de pesquisa e aplicação significativa de base fundamental para a sustentação de novas tecnologias e para a criação de aplicações mais ricas e interativas. Através da estruturação e organização de dados, a Web Semântica permite que diferentes sistemas e tecnologias se integrem de maneira mais eficaz, resultando em experiências de usuário aprimoradas, práticas e acessíveis (ISOTANI *et al.*, 2009).

Atualmente, seu uso é amplamente utilizado em diversas aplicações práticas e setores estratégicos, como saúde, educação e inteligência artificial, ao facilitar a criação de sistemas de dados interligados e interoperáveis. Com a evolução de padrões como RDF, OWL e SPARQL, o desenvolvimento de novas ferramentas e plataformas tem simplificado a implementação dessa tecnologia, impulsionando iniciativas como os *Linked Open Data (LOD)*. Um exemplo prático é o projeto *DBpedia*, que organiza informações estruturadas extraídas da Wikipédia, tornando-as acessíveis e reutilizáveis por diferentes sistemas. Além disso, a Web Semântica está cada vez mais integrada a tecnologias emergentes, como Inteligência Artificial (IA) e *Big Data*. Assistentes virtuais, como Siri e Alexa, e sistemas de busca semântica, como o Google, aplicam

conceitos semânticos para oferecer respostas mais precisas e contextualizadas, melhorando a recuperação da informação e otimizando a experiência do usuário (SOUZA; ALVARENGA, 2004; SEGUNDO, 2015; HITZLER, 2021).

No contexto educacional, conforme destacado no artigo *“Estado da Arte em Web Semântica e Web 2.0: Potencialidades e Tendências”* de Isotani *et al.* (2009), a Web Semântica foi incorporada ao conceito de Web 2.0, que prioriza a interação social e a colaboração. Essa convergência possibilitou a criação de ambientes de aprendizagem mais interativos, com sistemas que utilizam ontologias e metadados para classificar, organizar e recomendar conteúdos conforme o perfil de cada aluno. Plataformas educacionais semânticas são capazes de sugerir leituras e atividades personalizadas com base no desempenho e nas preferências de aprendizado dos usuários de forma mais eficiente e colaborativa.

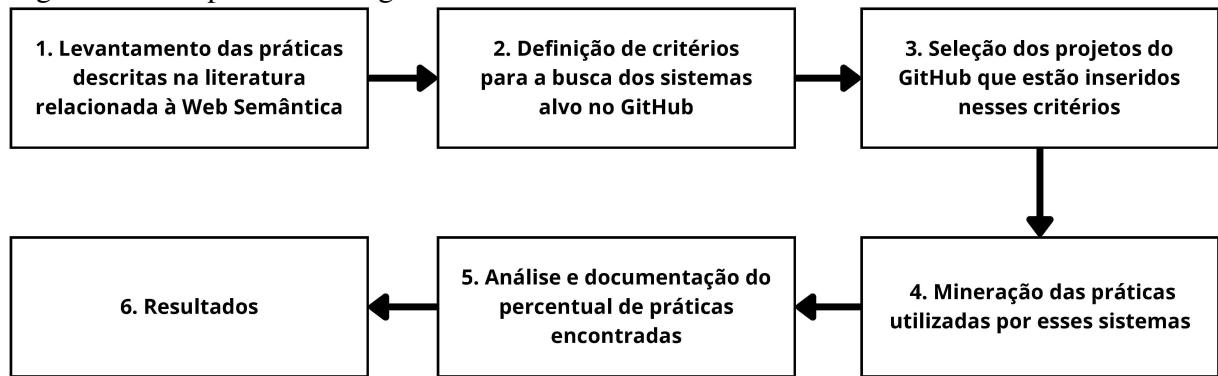
Diante das constantes inovações tecnológicas, a Web Semântica evolui como uma ferramenta crítica para enfrentar os desafios da organização e integração de informações na era digital. Seu papel vai além da mera interligação de dados, sendo um impulsionador para a criação de ambientes mais inteligentes, colaborativos e personalizados, contribuindo para a construção de uma internet mais conectada e orientada ao usuário.

### 3 METODOLOGIA

Este capítulo descreve os procedimentos metodológicos adotados para a realização deste estudo, cujo objetivo foi investigar as práticas da Web Semântica no cenário atual. A pesquisa foi conduzida em duas etapas principais: A busca dos projetos no *GitHub* que são relacionados as principais tecnologias da Web Semântica e a análise e mineração dos projetos encontrados no *GitHub* para verificar a adoção dessas tecnologias.

A metodologia proposta inclui as seguintes etapas: levantamento das práticas descritas na literatura relacionada à Web Semântica, definição de critérios para a busca dos sistemas alvo no *GitHub*, seleção dos projetos, mineração das práticas, análise do percentual das práticas encontradas e, por fim, apresentação dos resultados e sugestões para trabalhos futuros. A Figura 13 apresenta um diagrama de atividades com todas as etapas que serão descritas nas próximas seções.

Figura 13 – Etapas metodológicas



Fonte: Produzido pela autora.

A Seção 3.1 apresenta o levantamento das práticas relacionadas ao tema. Em seguida, a Seção 3.2 descreve os critérios definidos para a busca automatizada de projetos no *GitHub*. A seleção dos repositórios que atendem aos critérios estabelecidos é detalhada na Seção 3.3. A Seção 3.4 trata da mineração das práticas adotadas pelos projetos identificados. Já a Seção 3.5 discute a análise do percentual de ocorrência das práticas em cada projeto.

#### 3.1 Levantamento das práticas descritas na literatura relacionada à Web Semântica

A primeira etapa do estudo se dará com uma revisão bibliográfica. A revisão bibliográfica tem como objetivo principal explorar e analisar as fontes existentes na literatura

acadêmica e técnica relacionadas às principais tecnologias da Web Semântica, Resource Description Framework, Extensible Markup Language, SPARQL e Web Ontology Language. Durante esta etapa, foram examinados artigos científicos, livros, documentações oficiais e outros materiais pertinentes que tratam dessas tecnologias e sua aplicação no desenvolvimento de sistemas e aplicações. A revisão foi conduzida de forma manual, via ad hoc, com o intuito principal de obter uma visão geral e identificar quais são as principais ferramentas e tecnologias da Web Semântica. Essa abordagem proporciona uma compreensão inicial das características, recursos, benefícios e desafios associados a cada uma das tecnologias analisadas. Como resultado desta pesquisa, será apresentada uma análise para identificar as práticas da Web Semântica no cenário atual.

### 3.2 Definição de critérios para a busca dos sistemas alvo no *GitHub*.

Para garantir uma análise representativa e confiável, foram estabelecidos critérios específicos para a seleção dos projetos no *GitHub* relacionados à Web Semântica. A escolha dos indicadores listados na Tabela 6 se deu pelo fato de representarem as principais métricas disponibilizadas pela própria plataforma para avaliar repositórios de código aberto. Esses parâmetros são amplamente utilizados para mensurar a atividade, a popularidade e o nível de colaboração de projetos hospedados no *GitHub*. Dessa forma, optou-se por utilizar esses indicadores por serem os mais adequados e consistentes para refletir o engajamento da comunidade e a evolução dos projetos analisados.

Tabela 6 – Indicadores utilizados para análise de projetos no *GitHub*

| Indicador            | Descrição                                              |
|----------------------|--------------------------------------------------------|
| <b>Actions</b>       | Automação de processos e integração contínua.          |
| <b>Branches</b>      | Estrutura e organização do desenvolvimento.            |
| <b>Contributors</b>  | Diversidade e colaboração entre desenvolvedores.       |
| <b>Discussions</b>   | Interação e engajamento da comunidade.                 |
| <b>Fork</b>          | Reutilização e interesse por parte de outros usuários. |
| <b>Issues</b>        | Acompanhamento e gestão de problemas.                  |
| <b>Projects</b>      | Planejamento e organização de tarefas.                 |
| <b>Pull Requests</b> | Colaboração ativa no código do projeto.                |
| <b>Releases</b>      | Atualizações e melhorias contínuas.                    |
| <b>Star</b>          | Popularidade e reconhecimento pela comunidade.         |
| <b>Watch</b>         | Interesse em acompanhar atualizações do projeto        |

Fonte: Produzido pela autora.

Conforme os critérios definidos na Tabela 6, foram estabelecidos os principais indicadores utilizados para avaliar o interesse e a relevância dos projetos. Já a Tabela 7 apresenta os valores mínimos estabelecidos para cada um desses indicadores, que serviram como base para a filtragem dos repositórios analisados. Dessa forma, somente os projetos que atendiam aos limiares mínimos descritos na Tabela 7 foram considerados para as análises subsequentes.

Tabela 7 – Indicadores com limiar mínimo

| Indicador            | Limiar |
|----------------------|--------|
| <b>Watch</b>         | 11     |
| <b>Star</b>          | 10     |
| <b>Pull Requests</b> | 3      |
| <b>Discussions</b>   | 3      |
| <b>Projects</b>      | 3      |
| <b>Fork</b>          | 2      |
| <b>Contributors</b>  | 1      |
| <b>Branches</b>      | 1      |
| <b>Issues</b>        | 1      |
| <b>Actions</b>       | 1      |
| <b>Releases</b>      | 1      |

Fonte: Produzido pela autora.

### 3.3 Seleção dos projetos do *GitHub* que estão inseridos nesse critério

Inicialmente, a seleção dos projetos foi realizada manualmente por meio da aba de pesquisa do *GitHub*, com palavras-chave atrelados às tecnologias da Web Semântica. Tal abordagem exigiu a avaliação individual de cada repositório achado, verificando se eles eram relevantes e se encaixavam nos critérios definidos. Porém, devido ao grande número de projetos necessário para formar uma amostra significativa e ao elevado tempo gasto na pesquisa manual, optou-se por automatizar a busca, filtragem e análise de repositórios no *GitHub*. Para isso, foi desenvolvido um programa em *Python* que seguiu as seguintes etapas:

1. **Definição das palavras-chave:** Para uma busca mais ampla, as palavras-chave mantidas foram “Semantic Web”, “Web Semântica”, e suas variações gramaticais.
2. **Busca automatizada na Application Programming Interface (API) do GitHub:** O código em *Python* foi configurado para realizar buscas nos repositórios utilizando a API pública do *GitHub*, retornando informações conforme os limiares indicados na Tabela 7.
3. **Aplicação de filtros:** Os resultados foram filtrados com base nas palavras-chave e no número mínimo de *Star*, conforme o critério definido.

4. **Armazenamento dos resultados:** Os projetos que atenderam aos critérios foram organizados em uma tabela contendo informações detalhadas sobre cada repositório, facilitando a análise subsequente.

Ao invés de buscar todas as palavras-chave definidas anteriormente, optou-se por utilizar apenas as mais gerais. Essa escolha se deve à sua maior abrangência, à capacidade de retornar repositórios que mencionam os termos em diferentes campos e à flexibilidade de pesquisa em múltiplas áreas (nome, descrição, *readme* e tópicos). Caso a busca fosse realizada apenas por palavras-chave específicas associadas a cada tecnologia principal, o filtro se tornaria excessivamente restrito. Assim, projetos relevantes que utilizam tecnologias relacionadas à Web Semântica, mas que não foram explicitamente incluídas no conjunto de palavras-chave, poderiam ser excluídas dos resultados, comprometendo, portanto, a abrangência e a precisão da análise.

### 3.3.1 Automatização da seleção dos projetos

Para facilitar a coleta dos projetos relacionados à Web Semântica, foi desenvolvido um *script*<sup>1</sup> em *Python* executado no Google Colab, uma plataforma online oferecida pelo Google que permite a execução de código *Python* diretamente no navegador (Google, 2022). O script foi executado em um computador com sistema operacional Windows 10, processador Intel(R) Core(TM) i5, 8GB de memória RAM, sistema operacional de 64bits. A execução apresentou um tempo médio de aproximadamente 212s (cerca de 3,5 minutos), processando aproximadamente 3066 repositórios no total durante a coleta dos dados. A seguir, apresenta-se o código dividido por partes e suas respectivas explicações.

**Importação da biblioteca:** No Listing 3.3.1, a biblioteca “*requests*” foi utilizada para realizar requisições HTTP à API do *GitHub*.

1 `import requests`

Listing 3.3.1 – Importação de biblioteca

**Definição dos parâmetros de busca:** No Listing 3.3.2, as palavras-chave definidas em “*KEYWORDS*” são utilizadas para representar os termos mais amplos relacionados à Web Semântica, abrangendo diferentes formas de escrita, incluindo variações com e sem acento, a fim de garantir uma busca mais abrangente. Além disso, foi definido “*MIN\_STARS*”, um

<sup>1</sup> <https://colab.research.google.com/drive/1xTdRt3zUpAlaRa7XkuLoyy2Hz1a6Ecuq?usp=sharing>

número mínimo de *Star* como critério de popularidade para filtrar apenas os projetos mais relevantes. Esse limite foi determinado com base em testes iterativos, visando identificar o valor que retornasse os resultados mais representativos e de melhor qualidade.

Para realizar as requisições à API do *GitHub*, também é necessário fornecer um token de autenticação pessoal, o “*GITHUB\_TOKEN*”. Esse *token* é uma chave única gerada pelo próprio usuário diretamente em sua conta do *GitHub*, na área *Personal access tokens*. Ele permite autenticar as chamadas feitas à API, aumentando os limites de requisições e garantindo acesso a dados mais completos. Sem a utilização do *token*, a API impõe restrições no número de requisições permitidas por hora, o que pode comprometer a automação e a coleta eficiente dos dados.

```

1 KEYWORDS = ['"Semantic Web"', '"semantic web"', '"Web Semântica"', '"Web
2   Semantica"', '"web semântica"', '"web semantic"']
3 GITHUB_TOKEN = 'token'
```

Listing 3.3.2 – Configuração de parâmetros de busca

**Definição da função de busca dos repositórios:** O Listing 3.3.3 apresenta a função “*search\_github\_repositories*”, criada para pesquisar os repositórios via API com base nas palavras-chave especificadas no Listing 3.3.2. O parâmetro “*query*” representa o termo de busca e “*max\_pages*” define o número máximo de páginas a serem percorridas. A URL da requisição é montada com os filtros especificados, campos nome e descrição e ordenação por *Star*. O cabeçalho inclui o token de autenticação. Caso a resposta não seja bem-sucedida (*status* diferente de 200), a execução é interrompida. Os repositórios encontrados em cada página são extraídos do JSON de resposta e adicionados a lista geral. A função retorna todos os repositórios coletados após percorrer as páginas disponíveis.

```

1 def search_github_repositories(query, max_pages=10):
2     all_repositories = []
3     page = 1
4     while page <= max_pages:
5         url = f"https://api.github.com/search/repositories?q={query}+"
6         in:name,description&sort=stars&order=desc&per_page=200
7         &page={page}"
8         headers = {'Authorization': f'token {GITHUB_TOKEN}'}
9         response = requests.get(url, headers=headers)
10        if response.status_code != 200:
11            print(f"Erro: {response.status_code}")
12            break
13        repositories = response.json().get('items', [])
14        if not repositories:
15            break
16        all_repositories.extend(repositories)
17        page += 1
18    return all_repositories

```

Listing 3.3.3 – Função para busca de repositórios via API do *GitHub*

**Definição da função de filtragem por número mínimo de *Star*:** O Listing 3.3.4 apresenta o processo de análise de repositórios no *GitHub*, é comum que muitos projetos encontrados não sejam relevantes o suficiente para serem considerados na pesquisa. Para garantir um nível mínimo de popularidade e engajamento, foi implementado a função “`filter_by_stars`”, que tem como objetivo filtrar os repositórios que possuem no mínimo 10 *Star*.

A função recebe dois parâmetros: uma lista de repositórios (`repositories`) e um valor inteiro (`min_stars`), que representa o número mínimo de *Star* desejado. Em seguida, ela percorre essa lista e seleciona apenas os repositórios cujo campo “`stargazers_count`” seja maior ou igual ao valor definido como mínimo. Por fim, a função retorna apenas os repositórios que atendem ao critério mínimo de popularidade, reduzindo o ruído nos dados e possibilitando uma análise mais precisa e significativa.

```

1 def filter_by_stars(repositories, min_stars):
2     return [repo for repo in repositories if repo['stargazers_count'] >=
3             min_stars]

```

Listing 3.3.4 – Função para filtrar repositórios por número mínimo de *Star*

**Definição da função para obter métricas detalhadas de um repositório:** No Listing 3.3.5, a função “`get_repository_metrics`” recebe como parâmetro a URL de um

repositório no *GitHub*, o “repo\_url” e realiza requisições à API para coletar métricas detalhadas relacionadas a esse repositório. Inicialmente, define-se o cabeçalho HTTP com o *token* de autenticação pessoal “GITHUB\_TOKEN”, para garantir o acesso autorizado e maior limite de requisições. Em seguida, a função faz uma requisição GET para a URL do repositório para obter informações básicas no formato *JavaScript Object Notation* (JSON).

A partir dessa resposta, são extraídos os dados referentes ao número de bifurcações (*forks*) do repositório, o número de *Star* (*stargazers*) recebidas, entre outros, conforme definido na Tabela 6. Por fim, essas informações são armazenadas e retornadas no dicionário “metrics”.

```

1  def get_repository_metrics(repo_url):
2      headers = {
3          'Authorization': f'token {GITHUB_TOKEN}'
4      }
5      metrics = {}
6
7      repo_info = requests.get(repo_url, headers=headers).json()
8      metrics['forks'] = repo_info.get('forks_count', 0)
9      metrics['stars'] = repo_info.get('stargazers_count', 0)
10     metrics['watchers'] = repo_info.get('watchers_count', 0)
11     metrics['issues'] = repo_info.get('open_issues_count', 0)
12     metrics['languages'] = requests.get(repo_info['languages_url'],
13                                         headers=headers).json()
13     metrics['contributors'] = len(requests.get(repo_info['contributors_url'],
14                                     headers=headers).json())
14     metrics['branches'] =
15         len(requests.get(repo_info['branches_url'].replace('{/branch}', ''),
16                         headers=headers).json())
16     metrics['releases'] =
17         len(requests.get(repo_info['releases_url'].replace('{/id}', ''),
18                         headers=headers).json())
18     metrics['actions'] = len(requests.get(f'{repo_info['url']}/actions/runs',
19                                 headers=headers).get('workflow_runs', []))
20     metrics['discussions'] =
21         len(requests.get(f'{repo_info['url']}/discussions',
22                         headers=headers).json())
22     metrics['projects'] = len(requests.get(f'{repo_info['url']}/projects',
23                                         headers=headers).json())
23     metrics['pull_requests'] = len(requests.get(f'{repo_info['url']}/pulls',
24                                     headers=headers).json())
24
25
26     return metrics

```

Listing 3.3.5 – Função para obter métricas detalhadas de um repositório

**Busca na API do GitHub com base em palavras-chave:** No Listing 3.3.6 é

realizado a busca dos repositórios relacionados com a Web Semântica, foi implementado um laço de repetição que percorre a lista de palavras-chave “KEYWORDS”. Para cada termo presente nessa lista, é realizada uma chamada à função “`search_github_repositories`”, que se comunica com a API do *GitHub* e retorna uma lista de repositórios relacionados. Os resultados obtidos em cada iteração são adicionados na lista “`all_repositories`”, que passa a conter o conjunto completo de repositórios selecionados para todas as palavras-chave.

Essa abordagem permite selecionar os projetos que utilizam termos equivalentes (por exemplo, “*Semantic Web*” e “*Web Semântica*”) e, assim, obter uma amostra mais representativa. Embora isso possa gerar repositórios duplicados, essa duplicação é tratada posteriormente na etapa de remoção.

```

1 all_repositories = []
2 for keyword in KEYWORDS:
3     repositories = search_github_repositories(keyword, max_pages=10)
4     all_repositories.extend(repositories)

```

Listing 3.3.6 – Buscas iterativas na API do *GitHub* com base em palavras-chave

**Filtragem com base em *Star*:** No Listing 3.3.7 é realizado a chamada da função “`filter_by_stars`” e executa a filtragem dos repositórios coletados da API do *GitHub*, armazenados na lista “`all_repositories`” com base no número mínimo de *Star*. Este processo garante que apenas os projetos mais populares sejam mantidos para as etapas seguintes de análise. O resultado é armazenado na variável “`filtered_repositories`”.

```

1 filtered_repositories = filter_by_stars(all_repositories, MIN_STARS)

```

Listing 3.3.7 – Filtragem dos repositórios por número mínimo de *Star*

**Remoção dos repositórios duplicados:** O Listing 3.3.8 remove os repositórios repetidos que possam ter sido retornados mais de uma vez na busca. Para isso, os repositórios filtrados são convertidos em um dicionário, utilizando a URL como chave única e o próprio repositório como valor. Como dicionários não permitem chaves duplicadas, esse processo elimina automaticamente eventuais entradas com a mesma URL, o que garante que, se houver repositórios repetidos, apenas um deles será mantido. Em seguida, apenas os valores do dicionário, ou seja, os próprios repositórios, são extraídos e armazenados na variável “`unique_repositories`”, resultando em uma lista final sem redundâncias.

```

1 repo_dict = {}
2 for repo in filtered_repositories:
3     repo_dict[repo['url']] = repo
4
5 unique_repositories = repo_dict.values()

```

Listing 3.3.8 – Remoção dos repositórios duplicados

**Coleta as métricas detalhadas para cada repositório:** No Listing 3.3.9, após a etapa de filtragem, seleção e eliminação de projetos duplicados, e com as métricas devidamente estabelecidas, inicia-se o processo de coleta dos dados correspondentes para cada projeto. Para isso, foi implementado um laço de repetição que percorre todos os elementos da lista “unique\_repositories”. Para cada item, a função “get\_repository\_metrics” é chamada, recebendo como argumento a URL da API correspondente ao repositório. Em seguida, o dicionário original é atualizado com os dados obtidos, incorporando essas informações diretamente ao objeto, que é então armazenado na lista “repositories\_with\_metrics”.

```

1 repositories_with_metrics = []
2 for repo in unique_repositories:
3     metrics = get_repository_metrics(repo['url'])
4     repo.update(metrics)
5     repositories_with_metrics.append(repo)

```

Listing 3.3.9 – Coleta métricas detalhadas

**Exibição dos resultados:** Por fim, é realizado a exibição dos repositórios e suas respectivas métricas encontradas no Listing 3.3.10. Para isso, foi utilizado um laço de repetição que percorre todos os repositórios armazenados na lista “repositories\_with\_metrics”. A cada iteração, o índice do repositório e seus dados são acessados simultaneamente, permitindo numerar e exibir as informações de maneira sequencial e clara. Durante o processo, é impresso o nome do projeto, uma breve descrição, o link para o repositório no GitHub, os tópicos relacionados e as linguagens de programação utilizadas. Em seguida, são exibidas as principais métricas coletadas anteriormente. Dessa forma, essa visualização oferece uma visão detalhada e organizada de cada repositório, facilitando a análise e compreensão dos dados.

```

1 print(f"Qtd de projetos encontrados: {len(repositories_with_metrics)}\n")
2
3 for index, repo in enumerate(repositories_with_metrics, start=1):
4     print(
5         f"[{index}] "
6         f"Nome: {repo['name']} | "
7         f"Descrição: {repo.get('description', 'Sem descrição')} | "
8         f"URL: {repo['html_url']} | "
9         f"Tópicos: {', '.join(repo.get('topics', []))} | "
10        f"Linguagens: {', '.join(repo.get('languages', {}).keys())} | "
11        f"Estrelas: {repo['stars']} | "
12        f"Forks: {repo['forks']} | "
13        f"Watchers: {repo['watchers']} | "
14        f"Issues: {repo['issues']} | "
15        f"Pull Requests: {repo['pull_requests']} | "
16        f"Contribuidores: {repo['contributors']} | "
17        f"Branches: {repo['branches']} | "
18        f"Releases: {repo['releases']} | "
19        f"Actions: {repo['actions']} | "
20        f"Discussions: {repo['discussions']} | "
21        f"Projects: {repo['projects']}"
22    )

```

Listing 3.3.10 – Exibição dos resultados

### 3.4 Mineração das práticas da web semântica utilizadas por esses sistemas

Após a coleta dos repositórios por meio da API do *GitHub* executada na Seção 3.3, foi realizada a mineração das práticas utilizadas nos projetos selecionados. O objetivo principal foi avaliar a adesão da Web Semântica no cenário atual e identificar quais de suas tecnologias estão sendo mais utilizadas. Os critérios definidos para essa mineração foram essenciais para garantir a precisão, relevância e replicabilidade da análise. Eles foram estabelecidos com base em atributos diretamente relacionados às tecnologias dessa área, permitindo identificar se os projetos escolhidos utilizam seus conceitos e de que maneira. As tecnologias consideradas para compor esses critérios foram selecionadas a partir de uma revisão bibliográfica e da documentação oficial da W3C, que as reconhecem como as principais no contexto semântico da web.

A análise dos dados obtidos foi conduzida de forma quantitativa e descritiva, considerando a frequência de ocorrência das tecnologias nos repositórios selecionados. Também foram realizadas observações qualitativas pontuais para compreender de que modo essas tecnologias são aplicadas nos projetos, permitindo identificar padrões de uso e eventuais boas práticas.

- **Utilização das principais tecnologias:** RDF, OWL, SPARQL, JSON-LD, SHACL e

SKOS são amplamente reconhecidas como fundamentais na construção de aplicações semânticas. Detectá-las indica a adoção efetiva dessas práticas.

- **Presença de arquivos específicos:** Verificação da presença de arquivos com extensões ou nomes característicos das tecnologias, como .rdf, .owl, .rq, .jsonld, e termos como shacl, skos no nome dos arquivos. Isso facilita a verificação automatizada e permite uma análise objetiva e reproduzível.
- **Indicador de popularidade:** Foi analisada a recorrência das tecnologias da Web Semântica entre os repositórios com maior número de *Star*, considerando esse valor como um indicativo de relevância e aceitação pela comunidade.
- **Análise textual:** Leitura dos arquivos README.md, documentação e exemplos de uso para identificar menções explícitas ao uso de tecnologias da Web Semântica. Permite uma avaliação mais robusta, reduzindo falsos positivos e negativos.
- **Uso de bibliotecas ou vocabulários:** Inspeção do código-fonte em busca de bibliotecas específicas (como rdflib, owlready2, pyshacl) ou uso de prefixos como rdf :, owl :, sh :, skos :, entre outros. Aumento da capacidade generalização, esses critérios não estão vinculados a uma linguagem de programação específica, o que permite sua aplicação a projetos desenvolvidos com diferentes tecnologias.

Parte da mineração foi conduzida de forma automatizada por meio de *scripts Python*, que identificaram a presença desses elementos nos arquivos públicos dos repositórios. Quando necessário, a análise foi complementada com inspeção manual para garantir maior precisão.

### 3.4.1 *Filtragem por linguagens*

Para analisar a verificação da quantidade de repositórios que utilizam as linguagens da Web Semântica, foram utilizados os dados obtidos no Listing 3.3.9. Cada repositório armazenado em “repositories\_with\_metrics” contém um listagem das linguagens e para realizar uma filtragem com base nas palavras-chave dessas linguagens definidas foi executado o seguinte código no Listing 3.4.1:

```

1 web_semantic_keywords = [
2     "rdf", "owl", "web ontology language", "sparql", "shacl", "json-ld"
3 ]
4 provaveis_semantic_repos = []
5
6 for repo in repositories_with_metrics:
7     linguagens = repo.get('languages', {}).keys()
8     linguagens_lower = [lang.lower() for lang in linguagens]
9
10    if any(any(keyword in lang for keyword in web_semantic_keywords) for lang
11           in linguagens_lower):
12        provaveis_semantic_repos.append(repo)
13
14 print(f"Quantidade de repositórios com linguagens que indicam uso de
15       tecnologias semânticas: {len(provaveis_semantic_repos)}")
16 for r in provaveis_semantic_repos:
17     print(f"- {r['name']}")
```

Listing 3.4.1 – Filtragem por linguagem

Essa forma de filtragem mostrou-se limitada e resultou em um número reduzido de repositórios, uma vez que considera apenas as linguagens declaradas no repositório. Isso significa que, por exemplo, um projeto que utiliza RDF por meio da biblioteca *rdflib*, mas cuja linguagem principal é *Python*, não seria identificado por esse critério. Para contornar essa limitação, foi realizada uma nova abordagem de busca, desta vez baseada nas extensões de arquivos presentes nos repositórios, conforme apresentado na Seção 3.4.2.

### 3.4.2 *Filtragem por extensões de arquivos*

A identificação das tecnologias foi realizada por meio da verificação das extensões dos arquivos (.rdf, .owl, etc.), presentes nos repositórios selecionados e armazenados anteriormente em “`repositories_with_metrics`” no Listing 3.3.9, por se tratar de uma abordagem mais precisa e confiável. Essa estratégia permite identificar diretamente o tipo de conteúdo utilizado com base no formato do arquivo, reduzindo significativamente a ocorrência de falsos positivos, comuns em métodos baseados apenas na busca por palavras-chave ou termos no texto.

A função “`check_practice_in_repo()`” no Listing 3.4.2 foi utilizada para percorrer o conteúdo de cada repositório e verificar a presença dos seguintes indicadores:

- **RDF**: Arquivos com extensões .rdf, .ttl, .nt ou .n3.
- **OWL**: Arquivos com a extensão .owl.
- **SPARQL**: Arquivos com a extensão .rq ou nomes contendo a palavra sparql.

- **SHACL**: Arquivos ou diretórios contendo shacl no nome.
- **JSON-LD**: Arquivos com a extensão .jsonld ou nomes contendo json-ld.
- **XML-RDF**: Arquivos com extensão .xml, potencialmente estruturados no formato RDF/XML.

```

1 import requests
2
3 praticas_semanticas = ['RDF', 'OWL', 'SPARQL', 'SHACL', 'JSON-LD']
4 extensoes_rdf = ['.rdf', '.ttl', '.nt', '.n3']
5
6 def check_practice_in_repo(repo_contents_url, headers):
7     response = requests.get(repo_contents_url, headers=headers)
8     if response.status_code != 200:
9         return dict.fromkeys(praticas_semanticas, 0)
10
11     files = [item['name'].lower() for item in response.json()]
12
13     return {
14         'RDF': int(any(f.endswith(ext) for ext in extensoes_rdf for f in
15                         files)),
16         'OWL': int(any(f.endswith('.owl') for f in files)),
17         'SPARQL': int(any(f.endswith('.rq') or 'sparql' in f for f in files)),
18         'SHACL': int(any('shacl' in f for f in files)),
19         'JSON-LD': int(any(f.endswith('.jsonld') or 'json-ld' in f for f in
20                         files)),
21     }
22
23 semantic_repos = []
24
25 for repo in repositories_with_metrics:
26     repo_contents_url = repo['url'] + '/contents'
27     indicators = check_practice_in_repo(repo_contents_url, headers={
28         'Authorization': f'token {GITHUB_TOKEN}'})
29     repo.update(indicators)
30
31     if any(indicators[tec] == 1 for tec in praticas_semanticas):
32         semantic_repos.append(repo)

```

Listing 3.4.2 – Filtragem por extensão de arquivos

### 3.4.3 Coleta e consolidação dos dados dos repositórios

A partir dos repositórios identificados na etapa anterior, foi realizada a consolidação dos dados relevantes para as análises posteriores. Para cada repositório, foram coletadas métricas de popularidade, data da última atualização, linguagens de programação utilizadas, e os indica-

dores de uso das tecnologias da Web Semântica. O Listing 3.4.3 apresenta o código responsável por essa coleta.

```

1  dados_completos = []
2
3  for repo in semantic_repos:
4      dados_completos.append({
5          'Repositório': repo.get('name', 'Desconhecido'),
6          'Link': repo.get('html_url', 'N/A'),
7          'Estrelas': repo.get('stargazers_count', 0),
8          'Forks': repo.get('forks_count', 0),
9          'Watchers': repo.get('watchers_count', 0),
10         'Issues': repo.get('open_issues_count', 0),
11         'Contributors': len(requests.get(repo['contributors_url'],
12                             headers=headers).json()),
12         'Branches': len(requests.get(repo['branches_url'].replace('{/branch}', '',
13                             headers=headers).json()),
13         'releases': len(requests.get(repo['releases_url'].replace('{/id}', '',
14                             headers=headers).json()),
14         'Actions': len(requests.get(f'{repo["url"]}/actions/runs',
15                             headers=headers).json().get('workflow_runs', )),
15         'Discussions': len(requests.get(f'{repo["url"]}/discussions',
16                             headers=headers).json()),
16         'Projects': len(requests.get(f'{repo["url"]}/projects',
17                             headers=headers).json()),
17         'Pull_requests': len(requests.get(f'{repo["url"]}/pulls',
18                             headers=headers).json()),
18         'Última Atualização': repo.get('updated_at', 'N/A'),
19         'Linguagens': ", ".join(repo.get('languages', {}).keys()),
20         'Total Tecnologias': sum(repo.get(tec, 0) for tec in
21                             praticas_semanticas),
22         **{tec: repo.get(tec, 0) for tec in praticas_semanticas}
22     })

```

Listing 3.4.3 – Coleta consolidada dos dados dos repositórios para análise

### 3.5 Análise e documentação do percentual de práticas encontradas

Esta etapa teve como objetivo verificar a frequência de uso das práticas definidas anteriormente. A partir dos dados extraídos na etapa de mineração na Seção 3.4, foi criado um *DataFrame* contendo informações completas de cada repositório que apresentou ao menos uma tecnologia relacionada à Web Semântica. Esses dados incluem indicadores *booleanos* sobre o uso de cada tecnologia, bem como métricas de popularidade e características gerais dos repositórios. A Tabela gerada no Listing 3.5.1 reúne, para cada repositório:

- Indicadores de presença das tecnologias RDF, OWL, SPARQL, SHACL, JSON-LD e

## XML-RDF.

- Quantidade de cada critério definido na Tabela 6.
- Linguagens utilizadas.
- Data da última atualização.
- Total de tecnologias identificadas.

```

1 import pandas as pd
2
3 df_completo = pd.DataFrame(dados_completos)
4 df_completo['Última Atualização'] = pd.to_datetime(df_completo['Última
→ Atualização'], errors='coerce')
5 df_completo = df_completo.sort_values(by='Estrelas',
→ ascending=False).reset_index(drop=True)
6
7 df_completo.to_csv('tabela_completa_repos_semanticos.csv', index=False)
8 print("Arquivo salvo como 'tabela_completa_repos_semanticos.csv'")
```

Listing 3.5.1 – Criação da tabela e exportação CSV

### 3.5.1 Análise quantitativa das práticas da Web Semântica

Com base no dados gerados por meio do Listing 3.5.1, foi realizada a análise quantitativa da presença das tecnologias nos projetos. A estrutura *booleana* da tabela permitiu:

- Contabilizar quantos repositórios utilizam cada tecnologia.
- Calcular o número total de tecnologias por repositório.
- Ordenar os resultados conforme o grau de aderência tecnológica.
- Gerar visualizações que facilitam a interpretação.
- Agrupamento dos repositórios conforme o número de *Star*.
- Identificar as tecnologias predominantes nos repositórios mais populares.
- Analisar os padrões de adoção tecnológica relacionados ao engajamento da comunidade, tecnologias mais utilizadas entre os repositórios com mais *Star*.

Com base nesses dados, foi calculado a porcentagem de repositórios que adotam cada prática. O cálculo utilizou-se da fórmula representada na Figura 14. Posteriormente, esses percentuais foram organizados em um gráfico para facilitar a interpretação dos resultados.

Figura 14 – Fórmula para o cálculo percentual

$$\text{Percentual} = \left( \frac{\text{Número de repositórios que usam a tecnologia}}{\text{Total de repositórios encontrados}} \right) \times 100$$

Fonte: Produzido pela autora.

### 3.6 Perguntas de Pesquisa

Para orientar a análise quantitativa e a interpretação dos resultados, foram definidas as seguintes perguntas de pesquisa (QPs):

*QP<sub>1</sub>: Qual tecnologia da Web Semântica é a mais utilizada atualmente nos repositórios analisados?*

*QP<sub>2</sub>: Quantas tecnologias distintas são adotadas por projeto e como se distribui essa diversidade?*

*QP<sub>3</sub>: Quais linguagens de programação são mais utilizadas nos projetos de Web Semântica?*

*QP<sub>4</sub>: Os projetos de Web Semântica analisados apresentam atividade recente?*

*QP<sub>5</sub>: Como se relacionam a popularidade dos repositórios e o engajamento da comunidade nos projetos de Web Semântica?*

Essas questões de pesquisa exploram aspectos fundamentais da adoção da Web Semântica, oferecendo uma visão abrangente sobre as tecnologias mais utilizadas, a diversidade de práticas, as linguagens de programação predominantes, a atividade recente dos projetos e o nível de engajamento da comunidade.

## 4 RESULTADOS E DISCUSSÕES

Neste capítulo, são apresentados e discutidos os resultados obtidos a partir da análise quantitativa dos repositórios do *GitHub* relacionados às principais tecnologias da Web Semântica. A análise consistiu em um levantamento sistemático das características dos projetos por meio de uma abordagem exploratória e documental, utilizando mineração automatizada de dados com *scripts* desenvolvidos para identificar a presença de práticas semânticas nos repositórios. Essa abordagem permitiu mensurar a frequência de adoção das tecnologias semânticas e obter uma visão geral do cenário atual de utilização dessas práticas no desenvolvimento colaborativo.

### 4.1 Apresentação Geral busca dos projetos

A primeira análise realizada foi referente à quantidade de repositórios que, de fato, utilizam as tecnologias da Web Semântica. A partir da seleção dos projetos do *GitHub* na Seção 3.3, foram encontrados 85 repositórios que possuem alguma relação, e desses, após a mineração foram identificados 27 projetos que apresentavam evidências concretas do uso de pelo menos uma das tecnologias relacionadas, como RDF, OWL, SPARQL, SHACL, JSON-LD e SKOS, como demonstrado na Figura 15. Esse dado inicial já indica que, apesar do número de repositórios relacionados à temática da Web Semântica seja expressivo, nem todos aplicam efetivamente as tecnologias centrais na prática.

Figura 15 – Processos para a modelagem de filtragem



Fonte: Produzido pela autora.

Embora esses repositórios tenham sido inicialmente identificados, as análises apresentadas neste capítulo consideram apenas aqueles que realmente aplicam as práticas da Web Semântica, conforme os critérios definidos na Seção 3.4. A filtragem adicional, realizada por meio de mineração automatizada e inspeção dos arquivos, resultou num conjunto menor, porém mais fiel ao uso real dessas tecnologias. Cabe ressaltar também que, a identificação dos repositórios que utilizam o vocabulário SKOS foi realizada a partir dos projetos que já apresentavam evidências de uso do RDF. Isso ocorre porque SKOS não é um formato de serialização separado,

mas sim um vocabulário definido dentro do próprio modelo de RDF. Dessa forma, a presença dele só pode ser verificada por meio da análise dos arquivos RDF dos repositórios, pois não existe uma extensão de arquivo ou formato específico associado exclusivamente ao uso de SKOS.

## 4.2 Análise Quantitativa

Os resultados da análise quantitativa referente a adoção das tecnologias da Web Semântica nos repositórios selecionados são apresentados a seguir. As informações foram obtidas a partir da mineração e consolidação dos dados coletados, conforme descrito na Seção 3.4. São apresentados dados sobre a frequência de uso de cada tecnologia, o percentual de repositórios que adotam essas práticas e gráficos que facilitam a visualização e interpretação desses resultados. A discussão das comparações abordam os atributos citados na Seção 3.5.1. Os estudos realizados respondem a *QP1*: “*Qual tecnologia da Web Semântica é a mais utilizada no cenário atual?*”.

### 4.2.1 Frequência de adoção das tecnologias da Web Semântica

Nesta etapa, foram avaliados quais repositórios utilizam cada uma das tecnologias da Web Semântica. A partir da mineração dos dados obtidos na busca descrita na Seção 3.4, foi gerado a Tabela 8 que representa a presença (✓) ou ausência das tecnologias em cada projeto.

Tabela 8 – Repositórios e tecnologias utilizadas

| Repositório                                          | RDF | OWL | SPARQL | SHACL | JSON-LD | SKOS |
|------------------------------------------------------|-----|-----|--------|-------|---------|------|
| jena <sup>1</sup>                                    |     |     | ✓      | ✓     |         |      |
| swap <sup>2</sup>                                    | ✓   |     | ✓      |       |         |      |
| dita-rdf <sup>3</sup>                                | ✓   |     | ✓      |       |         |      |
| semantic_forms <sup>4</sup>                          | ✓   |     | ✓      |       |         | ✓    |
| Bibliographic-Ontology-BIBO <sup>5</sup>             | ✓   | ✓   |        |       |         |      |
| rdf.sh <sup>6</sup>                                  | ✓   |     |        |       |         |      |
| sweet <sup>7</sup>                                   | ✓   |     | ✓      |       |         |      |
| LinkedGeoData <sup>8</sup>                           |     |     | ✓      |       |         |      |
| sparql.anything <sup>9</sup>                         |     |     | ✓      |       |         |      |
| SemWeb2NL <sup>10</sup>                              |     | ✓   | ✓      |       |         |      |
| corese <sup>11</sup>                                 |     |     | ✓      |       |         |      |
| lisp_practical_semantic_web <sup>12</sup>            |     |     | ✓      |       |         |      |
| luposdate <sup>13</sup>                              |     |     | ✓      |       |         |      |
| jruby_semantic_web_examples <sup>14</sup>            | ✓   |     |        |       |         |      |
| SWObjects <sup>15</sup>                              |     |     | ✓      |       |         |      |
| huviz <sup>16</sup>                                  |     |     | ✓      |       |         |      |
| dcat-suite <sup>17</sup>                             | ✓   |     |        |       |         |      |
| RomanticWeb <sup>18</sup>                            |     | ✓   |        |       | ✓       |      |
| Scute <sup>19</sup>                                  | ✓   |     |        |       |         |      |
| talend4sw <sup>20</sup>                              |     |     | ✓      |       |         |      |
| Sports-Ontology <sup>21</sup>                        |     | ✓   |        |       |         |      |
| ripple <sup>22</sup>                                 |     | ✓   |        |       |         |      |
| COEUS <sup>23</sup>                                  |     | ✓   |        |       |         |      |
| restful-grounding <sup>24</sup>                      |     | ✓   |        |       |         |      |
| semantic <sup>25</sup>                               |     | ✓   |        |       |         |      |
| Relation-extraction-using-Semantic-Web <sup>26</sup> |     | ✓   |        |       |         |      |
| Curso2016-2017 <sup>27</sup>                         |     | ✓   |        |       |         |      |
| <b>Total</b>                                         | 9   | 2   | 13     | 1     | 1       | 1    |

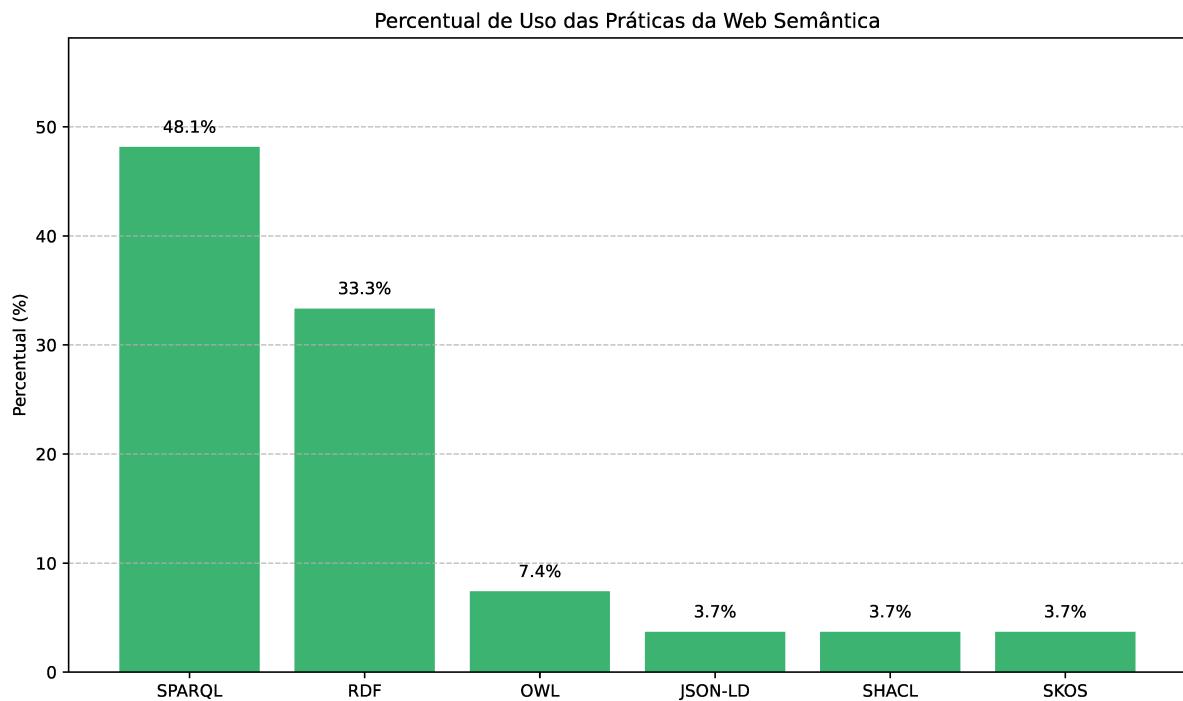
<sup>1</sup> <<https://github.com/apache/jena>>; <sup>2</sup> <<https://github.com/linkeddata/swap>>; <sup>3</sup> <<https://github.com/ColinMaudry/dita-rdf>>; <sup>4</sup> <[https://github.com/jmvanel/semantic\\_forms](https://github.com/jmvanel/semantic_forms)>; <sup>5</sup> <<https://github.com/structuredynamics/Bibliographic-Ontology-BIBO>>; <sup>6</sup> <<https://github.com/seebi/rdf.sh>>; <sup>7</sup> <<https://github.com/ESIPFed/sweet>>; <sup>8</sup> <<https://github.com/GeoKnow/LinkedGeoData>>; <sup>9</sup> <<https://github.com/SPARQL-Anything/sparql.anything>>; <sup>10</sup> <<https://github.com/AKSW/SemWeb2NL>>; <sup>11</sup> <<https://github.com/Wimmics/corese>>; <sup>12</sup> <[https://github.com/mark-watson/lisp\\_practical\\_semantic\\_web](https://github.com/mark-watson/lisp_practical_semantic_web)>; <sup>13</sup> <<https://github.com/luposdate/luposdate>>; <sup>14</sup> <[https://github.com/gigasquid/jruby\\_semantic\\_web\\_examples](https://github.com/gigasquid/jruby_semantic_web_examples)>; <sup>15</sup> <<https://github.com/ericprud/SWObjects>>; <sup>16</sup> <<https://github.com/smurp/huviz>>; <sup>17</sup> <<https://github.com/Scaseco/dcat-suite>>; <sup>18</sup> <<https://github.com/MakoLab/RomanticWeb>>; <sup>19</sup> <<https://github.com/danja/Scute>>; <sup>20</sup> <<https://github.com/fbelleau/talend4sw>>; <sup>21</sup> <<https://github.com/fbelleau/talend4sw>>; <sup>22</sup> <<https://github.com/joshsh/ripple>>; <sup>23</sup> <<https://github.com/bioinformatics-ua/COEUS>>; <sup>24</sup> <<https://github.com/otaviofff/restful-grounding>>; <sup>25</sup> <<https://github.com/superphy/semantic>>; <sup>26</sup> <<https://github.com/ekalgolas/Relation-extraction-using-Semantic-Web>>; <sup>27</sup> <<https://github.com/FacultadInformatica-LinkedData/Curso2016-2017>>.

Fonte: Produzido pela autora.

Para facilitar a compreensão da distribuição percentual das tecnologias nos projetos analisados, a Figura 16 apresenta um gráfico de barras com a proporção de repositórios que

adotam cada tecnologia considerada. Essa representação visual permite identificar com mais clareza quais tecnologias possuem maior adesão entre os projetos, além de evidenciar aquelas que aparecem de forma menos expressiva no conjunto analisado. A visualização complementa os dados da Tabela 8, reforçando as diferenças no nível de adoção e auxiliando na interpretação dos resultados quantitativos obtidos.

Figura 16 – Percentual de Uso das Práticas da Web Semântica



Fonte: Produzido pela autora.

**QP1.** A análise mostrou que SPARQL é a tecnologia mais utilizada, ressaltando sua relevância para consultas em grafos RDF. Observou-se também o uso significativo de RDF e OWL, evidenciando que essas tecnologias são aplicadas de forma complementar para estruturar, descrever e interrogar dados conforme as necessidades dos projetos.

A análise quantitativa evidencia que a tecnologia mais utilizada entre os projetos foi a SPARQL, presente em aproximadamente 48,1% dos repositórios, indicando um uso expressivo de consultas a dados semânticos estruturados. Esse resultado faz sentido, já que SPARQL é amplamente reconhecida como a linguagem padrão para consultas em grafos RDF e é bem suportada por diversas ferramentas consolidadas no ecossistema. Além disso, muitos projetos focam justamente na extração e manipulação de informações a partir de dados interligados, o que explica sua presença significativa. Isso mostra que, apesar da existência de outras práticas mais

recentes ou especializadas, a consulta a dados estruturados continua sendo uma das principais motivações para a adoção da Web Semântica nos projetos analisados.

Em seguida, destaca-se o RDF, com 33,3%, demonstrando que uma parcela significativa dos projetos ainda utiliza essa linguagem como base para representação de dados. Isso reforça o papel do RDF como um alicerce fundamental da Web Semântica, uma vez que ele define a estrutura básica dos grafos de dados sobre os quais as demais tecnologias operam. No entanto, a taxa de adoção relativamente menor em comparação ao SPARQL pode indicar que, em muitos casos, os desenvolvedores trabalham mais com a consulta e o consumo de dados RDF já existentes do que com a criação ou modelagem de novos conjuntos de dados RDF.

Tecnologias como SHACL, JSON-LD e SKOS apresentaram baixa adoção, estando presentes em apenas um ou dois repositórios, o que sugere um uso mais específico ou avançado dessas práticas. Esses resultados podem indicar que a validação de grafos (SHACL) ou a serialização voltada para interoperabilidade web (JSON-LD) ainda não se consolidou como práticas comuns na comunidade, seja por falta de demanda, pela maior complexidade de aplicação ou mesmo pela menor maturidade de ferramentas para esses padrões. Esse contexto evidencia uma preferência por tecnologias fundamentais da Web Semântica, como SPARQL e RDF, e uma menor difusão de ferramentas voltadas à validação SHACL ou serialização alternativa JSON-LD. A análise permite concluir que, embora esteja presente em diversos projetos, seu uso ainda se concentra em um conjunto restrito de tecnologias.

Além de identificar quais tecnologias são mais adotadas, também foi avaliado o número de tecnologias distintas empregadas em cada projeto. Para isso, foi somado a quantidade de tecnologias utilizadas por repositório, conforme indicado na última linha da Tabela 8. Essa contagem permite compreender o nível de complexidade semântica dos projetos, distinguindo aqueles que adotam apenas uma tecnologia daqueles que combinam múltiplas práticas da Web Semântica. **Respondendo a QP<sub>2</sub>:** “*Quantas tecnologias distintas da Web Semântica são adotadas por projeto?*”. Os resultados dessa contagem estão summarizados na Tabela 9, que apresenta a quantidade de projetos conforme o número de tecnologias utilizadas:

Tabela 9 – Distribuição dos projetos segundo o número de tecnologias utilizadas

| Nº de tecnologias por projeto | Nº de projetos |
|-------------------------------|----------------|
| 1                             | 19             |
| 2                             | 7              |
| ≥ 3                           | 1              |

Fonte: Produzido pela autora.

**QP<sub>2</sub>.** A maioria dos projetos analisados adota uma única tecnologia da Web Semântica, evidenciando uma adoção concentrada. Um número menor de projetos utiliza duas tecnologias, enquanto a adoção de três ou mais tecnologias é rara, indicando que a integração de múltiplas práticas ainda é limitada.

Esses resultados indicam que, embora haja um interesse crescente em explorar múltiplas tecnologias, a adoção combinada ainda é limitada na maioria dos projetos analisados. Essa tendência pode refletir desafios relacionados à integração, maturidade das ferramentas, ou mesmo o escopo dos projetos, que nem sempre demandam de múltiplas abordagens.

#### 4.2.2 *Características dos Projetos e Linguagens utilizadas*

**Respondendo a QP<sub>3</sub>:** “*Quais linguagens de programação são mais utilizadas nos projetos de Web Semântica?*”. A Tabela 10 apresenta a distribuição das linguagens de programação identificadas nos repositórios analisados.

Tabela 10 – Linguagens de programação utilizadas nos repositórios analisados

| Linguagem             | Qtd. de repositórios |
|-----------------------|----------------------|
| HTML                  | 12                   |
| Java                  | 11                   |
| Shell                 | 9                    |
| Makefile              | 7                    |
| Ruby                  | 7                    |
| Web Ontology Language | 6                    |
| XSLT                  | 6                    |
| JavaScript            | 5                    |
| Python                | 5                    |
| CSS                   | 5                    |
| Outras linguagens     | 19                   |

Fonte: Produzido pela autora.

**QP<sub>3</sub>.** A análise mostra que os projetos utilizam principalmente linguagens tradicionais do Desenvolvimento Web, como HTML e Java, combinadas com linguagens específicas para modelagem e transformação de dados semânticos, como Web Ontology Language e XSLT. Esse padrão reflete a integração entre tecnologias amplamente consolidadas e ferramentas voltadas ao contexto semântico.

Além disso, observa-se que um mesmo repositório pode empregar múltiplas lin-

guagens, motivo pelo qual a soma das ocorrências por linguagem excede o número total de repositórios analisados (27). Essa diversidade reforça o caráter interdisciplinar dos projetos relacionados à Web Semântica, que combinam diferentes abordagens para atender às suas necessidades.

#### **4.2.3 Atividade Recente dos Re却itórios**

**Respondendo a QP4:** “Os projetos de Web Semântica analisados apresentam atividade recente?”. A Tabela 11 apresenta os resultados da distribuição dos re却itórios selecionados conforme o ano da última atualização. Observa-se que 15 re却itórios foram atualizados em 2025 e outros 8 em 2024, representando aproximadamente 83% do total. Esse resultado indica que a adoção das tecnologias semânticas não é apenas teórica, mas também prática, com projetos mantidos e atualizados pela comunidade, ou seja, esses dados reforçam a relevância do critério de filtragem com base na atividade recente dos re却itórios, ajudando a selecionar projetos efetivamente mantidos e potencialmente mais representativos do uso atual da Web Semântica.

Tabela 11 – Distribuição dos re却itórios por ano da última atualização

| Ano da última atualização | Nº de projetos |
|---------------------------|----------------|
| 2025                      | 15             |
| 2024                      | 8              |
| 2023                      | 1              |
| 2022                      | 1              |
| 2019                      | 1              |
| 2017                      | 1              |

Fonte: Produzido pela autora.

**QP4.** A análise indica que a maioria dos projetos apresenta atividade recente, com atualizações concentradas principalmente nos anos de 2025 e 2024.

#### **4.2.4 Análise da Adoção Tecnológica nos re却itórios mais populares e Engajamento da Comunidade**

A popularidade e o engajamento comunitário dos projetos foram avaliadas por métricas como, o número de estrelas (*stars*), bifurcações (*forks*) e observadores (*watchers*) no *GitHub*. Essas métricas refletem tanto a visibilidade dos re却itórios quanto o interesse de desenvolvedores em acompanhar, reutilizar ou contribuir com os projetos. **Respondendo a**

**QP5:** “Como se relacionam a popularidade dos repositórios e o engajamento da comunidade nos projetos de Web Semântica?”. A Tabela 12 apresenta os repositórios analisados, ordenados de forma decrescente pelo número de estrelas.

Tabela 12 – Popularidade dos repositórios em termos de Stars, Forks e Watchers

| Repositório                            | Stars | Forks | Watchers |
|----------------------------------------|-------|-------|----------|
| jena                                   | 1184  | 662   | 1184     |
| sparql.anything                        | 251   | 15    | 251      |
| LinkedGeoData                          | 146   | 33    | 146      |
| sweet                                  | 131   | 34    | 131      |
| rdf.sh                                 | 123   | 9     | 123      |
| corese                                 | 111   | 28    | 111      |
| ripple                                 | 102   | 8     | 102      |
| Bibliographic-Ontology-BIBO            | 90    | 29    | 90       |
| semantic_forms                         | 76    | 21    | 76       |
| SemWeb2NL                              | 44    | 16    | 44       |
| lisp_practical_semantic_web            | 37    | 8     | 37       |
| swap                                   | 35    | 10    | 35       |
| COEUS                                  | 22    | 4     | 22       |
| dita-rdf                               | 21    | 5     | 21       |
| restful-grounding                      | 21    | 11    | 21       |
| luposdate                              | 19    | 11    | 19       |
| semantic                               | 18    | 3     | 18       |
| jruby_semantic_web_examples            | 17    | 4     | 17       |
| Relation-extraction-using-Semantic-Web | 17    | 4     | 17       |
| SWObjects                              | 15    | 4     | 15       |
| huviz                                  | 14    | 4     | 14       |
| dcat-suite                             | 13    | 2     | 13       |
| Curso2016-2017                         | 13    | 77    | 13       |
| Scute                                  | 12    | 2     | 12       |
| RomanticWeb                            | 12    | 9     | 12       |
| talend4sw                              | 11    | 9     | 11       |
| Sports-Ontology                        | 11    | 8     | 11       |

Fonte: Produzido pela autora.

**QP5.** A análise revela que os repositórios mais populares, medidos pelo número de *stars* no GitHub, também apresentam maior engajamento da comunidade, refletido pelo número elevado de *forks* e *watchers*.

A partir dela, observa-se que os projetos mais populares, como “jena”, “sparql.anything” e “LinkedGeoData”, também apresentam maior número de *forks* e *watchers*, o que exemplifica essa correlação e mostra que a popularidade está associada à visibilidade e à colaboração ativa dos desenvolvedores. Esses dados sugerem que o interesse da comunidade concentra-se em

projetos consolidados e amplamente utilizados.

A Tabela 13 aprofunda essa análise ao apresentar as tecnologias da Web Semântica utilizadas pelos cinco repositórios mais populares. Nota-se que tecnologias como SPARQL e RDF predominam nos projetos com maior número de estrelas, evidenciando que práticas como consultas semânticas e modelagem de dados em RDF estão fortemente associadas a projetos de maior impacto. Além disso, tecnologias mais recentes e especializadas, como SHACL, também aparecem entre os repositórios mais relevantes, indicando uma tendência de adoção de ferramentas de validação em contextos com maior visibilidade. Essa demonstração sugere que projetos que adotam essas tecnologias centrais tendem a atrair mais atenção e colaboração da comunidade.

Tabela 13 – Top 5 repositórios mais populares e suas respectivas tecnologias

| Repositório     | Tecnologias adotadas |
|-----------------|----------------------|
| jena            | SPARQL, SHACL        |
| sparql.anything | SPARQL               |
| LinkedGeoData   | SPARQL               |
| sweet           | RDF                  |
| rdf.sh          | RDF                  |

Fonte: Produzido pela autora.

### 4.3 Lições Aprendidas

Este trabalho proporcionou aprendizados significativos em termos técnicos e conceituais, uma vez que, inicialmente, o tema da Web Semântica e sua relevância no contexto atual da web eram pouco conhecidos. Essa limitação inicial dificultou o início do processo de escrita, exigindo um esforço adicional para a compreensão de conceitos básicos e avançados. Foi necessário realizar extensa pesquisa, incluindo leituras de artigos científicos, consultas a documentações oficiais e acompanhamento de vídeo-aulas, para consolidar uma visão geral sobre o funcionamento, as tecnologias e a importância da Web Semântica.

Apesar desse esforço, não foi possível compreender integralmente, na prática, a aplicação de todas as tecnologias envolvidas em projetos reais. Contudo, ficou evidenciada a relevância fundamental dessas tecnologias para a estruturação, interoperabilidade e acessibilidade dos dados na web, beneficiando iniciativas de dados abertos, aplicações de inteligência artificial e o desenvolvimento de ontologias. Também se observou que grafos de conhecimento auxiliam sistemas de IA a interpretar o significado dos dados e a responder a consultas complexas. A

revisão bibliográfica possibilitou compreender, ainda, a relação entre a Web Semântica, o HTML5 (com marcações semânticas que favorecem a acessibilidade) e o suporte a leitores de tela e agentes inteligentes, como exemplificado pelo projeto *schema.org*, já amplamente utilizado por mecanismos de busca e aplicações modernas.

A coleta de dados para análise revelou-se uma etapa particularmente desafiadora. Inicialmente, esperava-se encontrar um número expressivo de projetos no GitHub relacionados à Web Semântica. Entretanto, a realidade mostrou um cenário distinto, foi necessário desenvolver e refinar sucessivas vezes um *script* em *Python* para melhorar a precisão dos resultados. Ainda assim, o número de projetos encontrados com evidências concretas do uso das tecnologias permaneceu relativamente baixo, o que reforça a percepção de que, apesar de sua importância, a Web Semântica ainda é pouco difundida e utilizada ativamente. Essa constatação ajudou a dimensionar de forma mais realista as expectativas quanto ao volume de dados disponíveis e evidenciou a importância da flexibilidade para adaptar critérios e ajustar a metodologia ao longo do processo.

Outro desafio identificado foi a definição da estratégia para a busca no *GitHub*. Destacou-se, nesse processo, a necessidade de paciência e persistência diante das dificuldades práticas, como a ausência de padronização nas descrições dos repositórios e a necessidade de múltiplas rodadas de filtragem para obter um conjunto representativo. Foram consideradas diferentes abordagens para a busca, desde métodos manuais, utilizando a barra de pesquisa, até métodos automatizados, com consultas por palavras-chave ou por tópicos específicos. Cada abordagem apresentou vantagens e desvantagens: a busca manual revelou-se excessivamente trabalhosa e pouco produtiva, a automatizada com palavras-chave retornou resultados muito amplos e imprecisos, mesmo com filtro mínimo de *Star*, enquanto a busca por tópicos mostrou maior precisão, mas foi prejudicada pela baixa padronização dos metadados dos projetos. Essa etapa demandou mais tempo do que o previsto, reforçando a importância de um planejamento detalhado para futuros trabalhos com abordagens semelhantes.

Em termos gerais, a execução deste trabalho contribuiu para a aquisição de competências em organização, gestão do tempo e análise crítica, que poderão servir de base para projetos acadêmicos e profissionais futuros.

## 5 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho teve como objetivo principal investigar se as práticas da Web Semântica são utilizadas no cenário atual. Para isso, foi adotada uma abordagem metodológica sistemática, que combinou mineração automatizada de dados e inspeção manual de projetos hospedados no GitHub, possibilitando uma análise quantitativa dos repositórios que empregam essas tecnologias. Afim de garantir que os resultados refletissem o cenário atual e prático, a seleção dos projetos considerou os seguintes critérios:

- Um valor mínimo de *Actions, Branches, Contributors, Discussions, Fork, Issues, Projects, Pull Requests, Releases, Star e Watch*.
- Evidências concretas do uso de pelo menos uma das tecnologias principais: RDF, OWL, SPARQL, SHACL, JSON-LD ou SKOS;
- Atividade recente, a fim de refletir o estado atual da adoção dessas práticas;
- Relevantes para a temática, eliminando falsos positivos.

Os resultados obtidos evidenciam que, a Web Semântica é, sim, utilizada de forma prática e relevante, embora seu uso ainda esteja concentrado em nichos específicos e predominantemente em projetos voltados à organização e interoperabilidade de dados. Observou-se que a tecnologia mais empregada é o SPARQL, destacando-se como a principal linguagem para consultas a dados semânticos estruturados. Em seguida, destaca-se o RDF, reafirmando seu papel fundamental na representação dos dados. Tecnologias como OWL, SHACL, JSON-LD e SKOS surgem de forma pontual, indicando um uso mais especializado e menos disseminado.

Além disso, observou-se que projetos com maior popularidade (em termos de *stars, forks e watchers*) tendem a apresentar uma estrutura semântica mais consolidada, utilizando múltiplas tecnologias em conjunto, o que aponta para a importância da Web Semântica em soluções tecnicamente robustas e bem mantidas. Também se constatou que a maioria dos repositórios foi atualizada recentemente, em 2024 ou 2025, evidenciando que essa área permanece ativa atualmente e em constante evolução.

Uma contribuição significativa deste trabalho está na metodologia proposta, que se mostrou eficaz para identificar e classificar repositórios com base em evidências concretas do uso das tecnologias principais. Essa abordagem é facilmente replicável e pode ser adaptada para análises em outros contextos. Assim, este estudo não apenas confirma a presença e relevância das práticas da Web Semântica no desenvolvimento de software atual, como também serve como um caminho prático para estudos similares que demandem a triagem de grandes volumes de

projetos públicos.

Como proposta para trabalhos futuros, recomenda-se a ampliação do escopo da pesquisa para repositórios de outras plataformas, a investigação qualitativa das motivações para o uso das tecnologias semânticas e a análise do impacto dessas práticas na performance e na usabilidade dos sistemas desenvolvidos.

Com isso, pode-se concluir que a Web Semântica vem consolidando seu espaço como pilar fundamental para a construção de uma Web mais estruturada, interoperável e inteligente. Este estudo contribui para a literatura ao oferecer uma visão prática e atualizada das tendências de adoção da Web Semântica, fornecendo recursos para desenvolvedores e pesquisadores que desejam adotar essas tecnologias e explorar suas possibilidades.

## REFERÊNCIAS

- AGARWAL, V.; SASTRY, N. “way back then”: A data-driven view of 25+ years of web evolution. In: **Proceedings of the ACM Web Conference 2022**. [S.l.: s.n.], 2022. p. 3471–3479.
- AGUILAR, J. H. S.; ESTEVEZ, E. *et al.* Publication of linked open data—a systematic literature review for identifying problems and technical tools supporting the process. **Journal of Computer Science and Technology**, v. 23, n. 2, p. e16–e16, 2023.
- ALMEIDA, M. B. Uma introdução ao xml, sua utilização na internet e alguns conceitos complementares. **Ciência da informação**, SciELO Brasil, v. 31, p. 5–13, 2002.
- ANTONIOU, G.; HARMELEN, F. v. Web ontology language: Owl. **Handbook on ontologies**, Springer, p. 91–110, 2009.
- BECHHOFER, S.; HARMELEN, F. V.; HENDLER, J.; HORROCKS, I.; MCGUINNESS, D. L.; PATEL-SCHNEIDER, P. F.; STEIN, L. A. *et al.* Owl web ontology language reference. **W3C recommendation**, World Wide Web Consortium. [WWW document] <http://www.w3.org/TR/owl-ref>, v. 10, n. 2, p. 1–53, 2004.
- BERNERS-LEE, T. **Weaving the Web: The original design and ultimate destiny of the World Wide Web by its inventor**. [S.l.]: Harper San Francisco, 1999.
- BERNERS-LEE, T.; CAILLIAU, R.; GROFF, J.-F.; POLLERMANN, B. World-wide web: the information universe. **Internet Research**, MCB UP Ltd, v. 2, n. 1, p. 52–58, 1992.
- BERNERS-LEE, T.; CHEN, Y.; CHILTON, L.; CONNOLLY, D.; DHANARAJ, R.; HOLLENBACH, J.; LERER, A.; SHEETS, D. Tabulator: Exploring and analyzing linked data on the semantic web. In: ATHENS, GEORGIA. **Proceedings of the 3rd international semantic web user interaction workshop**. [S.l.], 2006. v. 2006, p. 159.
- BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. The semantic web. **Scientific american**, JSTOR, v. 284, n. 5, p. 34–43, 2001.
- BERNERS-LEE, T. J. The world-wide web. **Computer networks and ISDN systems**, Elsevier, v. 25, n. 4-5, p. 454–459, 1992.
- BIBI, N.; RANA, T. A.; MAQBOOL, A.; MIRZA, A.; IQBAL, Z.; KHAN, M. A.; ALHAISONI, M.; TARIQ, U.; DAMAŠEVIČIUS, R. Web semantics and ontologies-based framework for software component selection from online repositories. **International Journal of Web and Grid Services**, Inderscience Publishers (IEL), v. 19, n. 3, p. 318–349, 2023.
- BIZER, C.; HEATH, T.; BERNERS-LEE, T. Linked data—the story so far. In: **Linking the World's Information: Essays on Tim Berners-Lee's Invention of the World Wide Web**. [S.l.: s.n.], 2023. p. 115–143.
- BIZER, C.; HEATH, T.; IDEHEN, K.; BERNERS-LEE, T. Linked data on the web (ldow2008). In: **Proceedings of the 17th international conference on World Wide Web**. [S.l.: s.n.], 2008. p. 1265–1266.
- BREITMAN, K. K. **Web Semântica - A Internet do Futuro**. [S.l.]: Grupo GEN, 2005.

- CHAQFEH, M.; ASIM, R.; ALSHEBLI, B.; ZAFFAR, M. F.; RAHWAN, T.; ZAKI, Y. Towards a world wide web without digital inequality. **Proceedings of the National Academy of Sciences**, National Academy of Sciences, v. 120, n. 3, p. e2212649120, 2023.
- CONSORTIUM, W. W. W. **JSON-LD 1.1**. 2020. Disponível em: <<https://www.w3.org/TR/json-ld11/>>. Acesso em: 04 set. 2024.
- CONSORTIUM, W. W. W. **W3C History**. 2023. Disponível em: <<https://www.w3.org/about/history/>>. Acesso em: 06 dez. 2023.
- CONSORTIUM, W. W. W. *et al.* Extensible markup language (xml) 1.1. World Wide Web Consortium, 2006.
- CORMAN, J.; REUTTER, J. L.; SAVKOVIĆ, O. Semantics and validation of recursive shacl. In: SPRINGER. **The Semantic Web–ISWC 2018: 17th International Semantic Web Conference, Monterey, CA, USA, October 8–12, 2018, Proceedings, Part I 17**. [S.l.], 2018. p. 318–336.
- DECKER, S.; MITRA, P.; MELNIK, S. Framework for the semantic web: an rdf tutorial. **IEEE Internet Computing**, IEEE, v. 4, n. 6, p. 68–73, 2000.
- DIAS, T. D.; SANTOS, N. Web semântica: conceitos básicos e tecnologias associadas. **Cadernos do IME-Série Informática**, v. 14, p. 80–92, 2003.
- Dublin Core Metadata Initiative (DCMI). **Dublin Core Metadata Element Set, Version 1.1**. 2020. <<https://www.dublincore.org/specifications/dublin-core/dces/>>. Acesso em: 31 jul. 2025.
- DZIEKANIAK, G. V.; KIRINUS, J. B. Web semântica. 2004.
- EIS, D. **Introdução à Web Semântica: A inteligência da informação**. [S.l.]: Editora Casa do Código, 2017.
- FENSEL, D.; SIMSEK, U.; ANGELE, K.; HUAMAN, E.; KÄRLE, E.; PANASIUK, O.; TOMA, I.; UMBRICH, J.; WAHLER, A. **Knowledge graphs**. [S.l.]: Springer, 2020.
- Google. **Conheça o Colab**. 2022. <<https://colab.research.google.com/notebooks/welcome.ipynb?hl=pt-BR>>. Online; acessado em 7 de agosto de 2025.
- GOYAL, S.; SINGH, N. T.; RANI, N.; KAUR, A. Evolution and advancements of the world wide web: From concept to global phenomenon. In: IEEE. **2023 International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering (RMKMATE)**. [S.l.], 2023. p. 1–8.
- GRUBER, T. R. A translation approach to portable ontology specifications. **Knowledge acquisition**, Elsevier, v. 5, n. 2, p. 199–220, 1993.
- GUARINO, N.; OBERLE, D.; STAAB, S. What is an ontology? **Handbook on ontologies**, Springer, p. 1–17, 2009.
- GUTIERREZ, C.; HURTADO, C. A.; VAISMAN, A. Introducing time into rdf. **IEEE transactions on Knowledge and Data Engineering**, IEEE, v. 19, n. 2, p. 207–218, 2006.
- HEATH, T. Linked data: Evolving the web into a global data space. **Morgan & Claypool**, 2011.
- HENDLER, J. Is there an intelligent agent in your future? **Nature**, Springer Science and Business Media LLC, v. 11, 1999.

- HENDLER, J. Agents and the semantic web. **IEEE Intelligent systems**, IEEE, v. 16, n. 2, p. 30–37, 2001.
- HITZLER, P. A review of the semantic web field. **Communications of the ACM**, ACM New York, NY, USA, v. 64, n. 2, p. 76–83, 2021.
- IANNELLA, R.; WAUGH, A. **Metadata: enabling the Internet**. [S.l.]: DSTC Pty Limited, 1997.
- ISOTANI, S.; MIZOGUCHI, R.; BITTENCOURT, I. I.; COSTA, E. Estado da arte em web semântica e web 2.0: potencialidades e tendências da nova geração de ambientes de ensino na internet. **Revista brasileira de informática na educação**, v. 17, n. 01, p. 30, 2009.
- LAWSON, B.; SHARP, R. **Introducing html5**. [S.l.]: New Riders, 2011.
- MANOLA, F.; MILLER, E.; MCBRIDE, B. *et al.* Rdf primer. **W3C recommendation**, Citeseer, v. 10, n. 1-107, p. 6, 2004.
- MCBRIDE, B. The resource description framework (rdf) and its vocabulary description language rdfs. In: **Handbook on ontologies**. [S.l.]: Springer, 2004. p. 51–65.
- MCGUINNESS, D. L.; HARMELEN, F. V. *et al.* Owl web ontology language overview. **W3C recommendation**, v. 10, n. 10, p. 2004, 2004.
- MILES, A.; BECHHOFER, S. **SKOS Simple Knowledge Organization System Reference**. United States: World Wide Web Consortium, 2009. (W3C Recommendation).
- MILES, A.; MATTHEWS, B.; WILSON, M.; BRICKLEY, D. Skos core: simple knowledge organisation for the web. In: DUBLIN CORE METADATA INITIATIVE. **Proceedings of the International Conference on Dublin Core and Metadata Applications**. [S.l.], 2005.
- Ontotext. **What Is SHACL?** 2025. <<https://www.ontotext.com/knowledgehub/fundamentals/what-is-shacl/>>. Online; acessado em 7 de agosto de 2025.
- PÉREZ, J.; ARENAS, M.; GUTIERREZ, C. Semantics and complexity of sparql. **ACM Transactions on Database Systems (TODS)**, ACM New York, NY, USA, v. 34, n. 3, p. 1–45, 2009.
- PILGRIM, M. **HTML5: up and running: dive into the future of web development**. [S.l.]: "O'Reilly Media, Inc.", 2010.
- ŘEZNÍK, T.; RAES, L.; STOTT, A.; LATHOUWER, B. D.; PEREGO, A.; CHARVÁT, K.; KAFKA, Š. Improving the documentation and findability of data services and repositories: A review of (meta) data management approaches. **Computers & Geosciences**, Elsevier, v. 169, p. 105194, 2022.
- Schema.org. **Schema.org – Official Documentation**. 2024. <<https://schema.org>>. Site oficial com descrição dos tipos, propriedades e guias de uso em JSON-LD, RDFa e Microdata. Acesso em: 31 jul. 2025.
- SEGARAN, T.; EVANS, C.; TAYLOR, J. **Programming the semantic web: Build flexible applications with graph data**. [S.l.]: "O'Reilly Media, Inc.", 2009.

SEGUNDO, J. E. S. Web semântica, dados ligados e dados abertos: uma visão dos desafios do brasil frente às iniciativas internacionais. **Tendências da Pesquisa Brasileira em Ciência da Informação**, v. 8, n. 2, 2015.

SILVA, F. D. F. Uma nova abordagem de mineracao de repositÓrios de software utilizando ferramentas da web semântica. PUC-Rio, n. 4916, 2015.

SOUZA, R. R.; ALVARENGA, L. A web semântica e suas contribuições para a ciência da informação. **Ciência da Informação**, SciELO Brasil, v. 33, p. 132–141, 2004.

STĂNESCU, G.; OPREA, S.-V. Recent trends and insights in semantic web and ontology-driven knowledge representation across disciplines using topic modeling. **Electronics**, MDPI AG, v. 14, n. 7, p. 1313, 2025.

TAYE, M. M. Understanding semantic web and ontologies: Theory and applications. **arXiv preprint arXiv:1006.4567**, 2010.

W3C. **Shapes Constraint Language (SHACL)**: W3c recommendation. 2017. Disponível em: <<https://www.w3.org/TR/shacl/#introduction>>.

Wordlift. **Linked data**. 2025. <<https://wordlift.io/blog/en/entity/linked-data/>>. Online; acessado em 7 de agosto de 2025.