



**INSTITUTO FEDERAL  
SANTA CATARINA**

**CÂMPUS FLORIANÓPOLIS  
DEPARTAMENTO ACADÊMICO DE SAÚDE E SERVIÇO  
CURSO SUPERIOR DE TECNOLOGIA EM GESTÃO DA  
TECNOLOGIA DA INFORMAÇÃO**

**Alexandre Andrade Minichiello**

**Protótipo de aplicação web para  
dados abertos conectados**

**Florianópolis - SC  
2017**

Ficha de identificação da obra elaborada pelo autor

Minichiello, Alexandre Andrade  
Protótipo de aplicação web para dados abertos conectados  
/ Alexandre Andrade Minichiello ; orientação de Herval  
Daminelli; coorientação de Antônio Pereira Cândido.  
- Florianópolis, SC, 2017.  
76 p.

Trabalho de Conclusão de Curso (TCC) - Instituto Federal  
de Santa Catarina, Câmpus Florianópolis. CST  
em Gestão de TI. Departamento Acadêmico de Saúde  
e Serviços.  
Inclui Referências.

1. Web Semântica. 2. Dados Conectados. 3. Aplicação  
Web. I. Daminelli, Herval. II. Pereira Cândido, Antônio.  
III. Instituto Federal de Santa Catarina. Departamento  
Acadêmico de Saúde e Serviços. IV. Título.

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA  
CATARINA  
DEPARTAMENTO ACADÊMICO DE SAÚDE E SERVIÇOS  
CURSO SUPERIOR DE TECNOLOGIA EM GESTÃO DA TECNOLOGIA DA  
INFORMAÇÃO**

**ALEXANDRE ANDRADE MINICHIELLO**

**Protótipo de aplicação web para dados abertos conectados**

Trabalho de Conclusão de Curso submetido ao Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina como parte dos requisitos para obtenção do título de Tecnólogo em Gestão da Tecnologia da Informação.

Professor Orientador:

Prof. Herval Daminelli, Esp.

Professor Coorientador:

Prof. Antônio Pereira Cândido, Doutor.

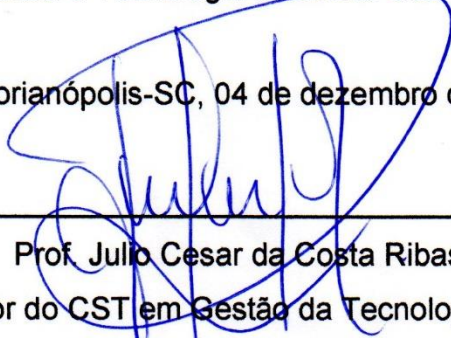
**FLORIANÓPOLIS - SC  
NOVEMBRO/2017**

## Protótipo de aplicação web para dados abertos conectados

**ALEXANDRE ANDRADE MINICHIELLO**

Este trabalho foi julgado adequado para obtenção do Título de Tecnólogo em Gestão da Tecnologia da Informação e aprovado na sua forma final pela banca examinadora do Curso Superior de Tecnologia em Gestão da Tecnologia da Informação do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

Florianópolis-SC, 04 de dezembro de 2017.



---

Prof. Julio Cesar da Costa Ribas, Dr.

Coordenador do CST em Gestão da Tecnologia da Informação  
Instituto Federal de Santa Catarina

Banca Examinadora:



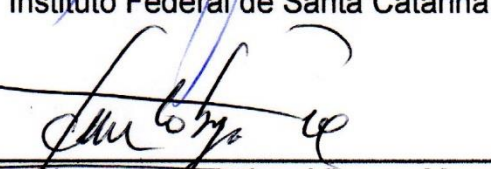
---

Prof. Herval Daminelli, Esp  
Orientador  
Instituto Federal de Santa Catarina



---

Glauco Cardozo, Me  
Instituto Federal de Santa Catarina



---

Cleverson Tabajara Vianna, Me  
Instituto Federal de Santa Catarina

*Dedico este trabalho a minha família, amigos, orientado, coorientador e  
a todos que ajudaram nesta luta.*

## **AGRADECIMENTOS**

Agradeço primeiramente aos meus pais, ao meu irmão e ao demais familiares por terem me incentivado durante toda esta jornada sem perderem a fé ou esperança. Ao meu orientador, professor Herval Daminelli, que sempre esteve ajudando nesta trajetória acadêmica, incentivando nos momentos difíceis e tendo paciência durante esta caminhada. Ao meu coorientador, professor Antônio Pereira Cândido, por sempre motivar e mostrar o melhor caminho rumo ao conhecimento.

*“A web não está concluída, é apenas a ponta do iceberg. As novas mudanças irão balançar o mundo ainda mais.”*  
*Tim Berners-Lee, Criador da World Wide Web.*

## RESUMO

O presente trabalho apresenta o desenvolvimento de um protótipo para uma aplicação web voltada para a temática de dados abertos conectados, sendo seu desenvolvimento, na maior parte, focado na linguagem em PHP e suas bases de dados em RDF e MySQL. O projeto inicia introduzindo os dados conectados, cuja conceituação é explanada para uma melhor compreensão da temática e dos assuntos relacionados, auxiliando no desenvolvimento do protótipo. A tecnologia e as ferramentas implementadas no projeto têm como objetivo principal dar o suporte ao desenvolvimento do protótipo.

Palavras-chave: Web Semântica. Dados conectados. Aplicação Web. Protótipo.



## **ABSTRACT**

The present work presents the development of a prototype for a web application focused on the subject of open data connected, being its main development in the language in PHP and its databases in RDF and MySQL. The initiative project exposing as concepts related to the area of connected data where they are explained for a better understanding of the thematic and related subjects, so that the prototype development was done. The technology and tools implemented in the project are based in order to support the development of the prototype.

Key-words: Semantic Web. Link data. Web Application. Prototype.

## LISTA DE ILUSTRAÇÕES

|  |    |
|--|----|
| Figura 1 - Arquitetura da Web Semântica .....                            | 19 |
| Figura 2 – Nuvem de datasets Link Data .....                             | 21 |
| Figura 3 - Etapas do sistema de 5 estrelas .....                         | 22 |
| Figura 4 - Representação de grafos RDF .....                             | 26 |
| Figura 5 - Representação RDF/XML - Informações DBpedia .....             | 26 |
| Figura 6 - Consulta SPARQL.....  | 27 |
| Figura 7 - Utilização do Where e Select.....                             | 28 |
| Figura 8 - Código XML.....   | 36 |
| Figura 9 - Tags HTML.....  | 37 |
| Figura 10 - Código utilizando heranças e seções .....                    | 41 |
| Figura 11 - Arquitetura de fornecimento de dados DBpedia atual .....     | 46 |
| Figura 12 - Site da Wikidata.....  | 47 |
| Figura 13 - Caso de Uso do Protótipo.....                                | 50 |
| Figura 14 - Diagrama de classe do Protótipo .....                        | 55 |
| Figura 15 - Página Inicial .....   | 56 |
| Figura 16 - Acesso Sistema.....  | 57 |
| Figura 17 - Tela de Cadastro .....                                       | 58 |
| Figura 18 - Sistema de Busca.....  | 59 |
| Figura 19 - Tela do Mapa e dados geográficos.....                        | 60 |
| Figura 20 - Classe User criada pelo Laravel no protótipo .....           | 61 |
| Figura 21 - Classe responsável pela criação das tabelas do Usuário ..... | 62 |
| Figura 22 - Classe controle do Registro do Usuário.....                  | 63 |
| Figura 23 - Código da tela de Login.....                                 | 64 |
| Figura 24 - Código Javascript do Mapa interativo .....                   | 65 |
| Figura 25 - Código para consulta das informações do Mapa .....           | 67 |
| Figura 26 - Classe teste do PHPUnit .....                                | 68 |
| Figura 27 - Tela com os erros apontados pelo PHPUnit .....               | 69 |
| Figura 28 -Tela informando que o teste deu certo. ....                   | 70 |

## LISTA DE QUADROS

|   |    |
|---|----|
| Quadro 1 - Sistema de 5 estrelas custo e benefícios ..... | 24 |
| Quadro 2 - Lista de API .....                             | 48 |
| Quadro 3 - Requisitos não funcionais .....                | 48 |
| Quadro 4 - Requisitos funcionais .....                    | 49 |

## LISTA DE ABREVIATURAS E SIGLAS

|               |   |
|---------------|---|
| <b>API</b>    | <i>Application Programming Interface</i> (Interface de Programação de Aplicações) |
| <b>CRUD</b>   | <i>Create, Read, Update e Delete</i> (Criar, Ler, Atualizar, Deletar)             |
| <b>CSS</b>    | <i>Cascading Style Sheets</i>   |
| <b>CSV</b>    | <i>Comma-separated values</i>   |
| <b>HTML</b>   | <i>HyperText Markup Language</i> (Linguagem de Marcação de Hipertexto)            |
| <b>JSON</b>   | <i>JavaScript Object Notation</i>   |
| <b>LOD</b>    | <i>Link Open Data</i> (Dados Abertos Conectados)                                  |
| <b>MVC</b>    | <i>Model-view-controller</i> (Modelo-visão-controlador)                           |
| <b>NS</b>     | <i>Namespace</i>  |
| <b>OWL</b>    | <i>Web Ontology Language</i>  |
| <b>PDF</b>    | <i>Portable Document Format</i>   |
| <b>PHP</b>    | <i>Personal Home Page</i>   |
| <b>POO</b>    | Programação orientada a objetos   |
| <b>RDF</b>    | <i>Resource Description Framework</i>   |
| <b>RDFS</b>   | <i>Resource Description Framework Schema</i>                                      |
| <b>SIOP</b>   | Sistema Integrado de Planejamento e Orçamento                                     |
| <b>SQL</b>    | <i>Structured Query Language</i> (Linguagem de Consulta Estruturada)              |
| <b>SPARQL</b> | SPARQL Protocol and RDF Query Language  |
| <b>UML</b>    | <i>Unified Modeling Language</i> (Linguagem de Modelagem Unificada)               |
| <b>URI</b>    | <i>Uniform Resource Identifier</i> (Identificador Uniforme de Recurso)            |
| <b>XML</b>    | <i>Extensible Markup Language</i>   |
| <b>W3C</b>    | <i>World Wide Web Consortium</i>  |

## SUMÁRIO

|   |    |
|---|----|
| 1. INTRODUÇÃO .....                           | 15 |
| 1.1. Justificativa.....                       | 15 |
| 1.2. Definição do Problema .....              | 16 |
| 1.3. Objetivos .....                          | 16 |
| 1.3.1. Objetivo Geral.....                    | 16 |
| 1.3.2. Objetivos Específicos .....            | 17 |
| 1.4. Estrutura do Trabalho.....               | 17 |
| 2. REVISÃO DA LITERATURA .....                | 18 |
| 2.1. Web Semântica .....                      | 18 |
| 2.2. Dados interligados.....                  | 20 |
| 2.3. RDF.....                                 | 24 |
| 2.4. SPARQL.....                              | 27 |
| 2.5. Ontologia.....                           | 28 |
| 2.6. OWL.....                                 | 30 |
| 2.7. Análise de Requisito Não Funcional ..... | 31 |
| 2.8. Análise de Requisito Funcional .....     | 32 |
| 2.9. Linguagem de Programação .....           | 33 |
| 2.10. Usabilidade .....                       | 34 |
| 2.11. XML.....                                | 35 |
| 2.12. HTML .....                              | 36 |
| 2.13. PHP.....                                | 37 |
| 2.14. CSS.....                                | 38 |
| 2.15. JavaScript .....                        | 39 |
| 2.16. Framework .....                         | 40 |
| 2.17. Laravel .....                           | 41 |

|  |    |
|--|----|
| 3. PROCEDIMENTOS METODOLÓGICOS .....                         | 43 |
| 4. DESENVOLVIMENTO .....                                     | 45 |
| 4.1. Ferramentas utilizadas .....                            | 45 |
| 4.1.1. Astah .....   | 45 |
| 4.1.2. Sublime Text .....                                    | 45 |
| 4.1.3. Pingendo .....  | 45 |
| 4.2. Base de dados em RDF .....                              | 45 |
| 4.2.1. DBpedia.....  | 46 |
| 4.2.2. Wikidata.....   | 47 |
| 4.2.3. SIOP.....   | 47 |
| 4.3. API para realização de busca RDF .....                  | 48 |
| 4.4. Análise de Requisito Não Funcional .....                | 48 |
| 4.5. Análise de Requisito Funcional .....                    | 49 |
| 4.6. Caso de Uso .....                                       | 49 |
| 4.6.1. Documentação do caso de uso .....                     | 50 |
| 4.7. Diagrama de classe.....                                 | 54 |
| 4.8. Desenvolvimento do Protótipo .....                      | 55 |
| 4.8.1. Layout da aplicação.....                              | 55 |
| 4.8.2. Programação utilizando Laravel .....                  | 60 |
| 4.8.3. Programação da consulta SPARQL.....                   | 66 |
| 4.9. Teste da aplicação .....                                | 68 |
| 4.10. Implementações Futuras .....                           | 70 |
| 4.10.1. Implementações de uma Sistema Geográfico (SIG) ..... | 70 |
| 4.10.2. Busca aprimorada .....                               | 71 |
| 4.10.3. Sistema para baixar os arquivos.....                 | 71 |
| 5. CONCLUSÕES .....  | 72 |
| Referências .....  | 74 |

## 1. INTRODUÇÃO

Com o crescente aumento das informações na web, sentiu-se a necessidade de acesso à diversas formas para a estruturação e publicação de dados na web, de forma que facilitasse a busca pela informação. Neste cenário, foi criado o conceito de dados conectados para trabalhar com a estruturação e a publicação desses dados de forma que as pessoas pudessem utilizar estas informações em suas atividades.

Um protótipo de aplicação web foi sugerido para trabalhar com os recursos ligados ao conceito de dados conectados, de forma a buscar informações em diversas bases em RDF (Resource Description Framework) e oferecer ao usuário estas informações num formato mais fácil e compreensível.

### 1.1. Justificativa

O Link Open Data (dados abertos conectados) é o conjunto de práticas introduzidas por Tim Berners-Lee para publicação e estruturação de dados na Web, de tal forma que dados fiquem interligados entre si. Este conjunto de práticas é representado pelo Resource Description Framework (RDF).

Link Open Data possui 3 princípios básicos a serem seguidos: deve ser aberto, para oferecer acesso a diversas aplicações; modular, para não ser necessário o planejamento prévio na ligação com outros dados, e escalável, pois a entrada de novos dados é facilitada caso já existam dados em RDF.

A utilização de LOD (Link Open Data) vem crescendo a cada ano, principalmente, por parte de instituições do governo, onde estão mantidos diversos dados de interesse da população. Some-se a isso, também, o aumento do número de aplicações para publicação e compartilhamento de Datasets (conjunto de dados organizado em forma de tabela).

Em virtude deste cenário, pretende-se construir o protótipo de uma aplicação web que possa gerenciar e mostrar, de forma mais compreensível, os dados que são compartilhados e publicados utilizando as práticas de LOD, para que essas informações não fiquem restritas apenas a uma parcela menor da população, como acontece

atualmente, onde os dados ficam disponíveis e são utilizados, majoritariamente, por repartições de órgãos públicos, governamentais e institucionais.

## **1.2. Definição do Problema**

Por ser a web o maior meio de disseminação de informação e o uso do LOD estar crescendo significativamente, mas com pouca visibilidade para os usuários que não estão ligados diretamente a área de informática, mas utilizam dados em LOD, faz-se necessário um mecanismo que permita a disseminação, aquisição e acesso a determinadas informações de forma mais objetiva, direta e simplificada. Desta forma, uma parcela cada vez maior de usuários pode dispor destes dados sem a complexidade e dificuldade que o atual sistema apresenta.

Como exemplo, podemos citar o problema e a dificuldade encontrados por diversos setores governamentais que necessitam acesso padronizado a informações relacionadas a estatísticas e dados, mas que, pelos meios convencionais, levando-se em consideração a forma restrita como o atual sistema trata estes dados, não são utilizados ou, quando muito, são utilizados em baixa escala.

Como tornar os dados abertos interligados (LOD) mais visíveis, compreensíveis e de fácil acesso para o público em geral? Esta é a questão que irá nortear o desenvolvimento desta pesquisa.

## **1.3. Objetivos**

Para o desenvolvimento deste trabalho foram definidos os seguintes objetivos: geral e específicos.

### **1.3.1. Objetivo Geral**

Desenvolver um protótipo de aplicação web para a extração e demonstração das informações das bases RDF.



### 1.3.2. Objetivos Específicos

- Conceituar Dados Abertos Conectados.
- Conceituar Web semântica.
- Desenvolver o protótipo do código-fonte da aplicação.
- Testar a aplicação.

### 1.4. Estrutura do Trabalho

Este trabalho está estruturado em cinco capítulos, sendo o primeiro a introdução do projeto, que contempla a justificativa, problema e os objetivos (principal e específicos) que tem a função de fazer a abertura do projeto, e expor os motivos e planos para o mesmo.

O segundo capítulo foca na revisão bibliográfica e é voltado para a parte de conceituação dos temas abordados, assim como a explicação sobre a importância dos mesmos.

O terceiro capítulo abrange os procedimentos metodológicos que serão utilizados para a realização do desenvolvimento, como a definição do tipo do projeto ou como será abordada a pesquisa.

O quarto capítulo trata do desenvolvimento do protótipo, que se concentra na execução prática de todo o planejamento do trabalho, descrevendo, de forma sucinta, o que foi feito e explicitando algumas das etapas do desenvolvimento.

O último capítulo contempla a conclusão, que traz os resultados do trabalho feito, permitindo avaliar aquilo que foi produzido no decorrer da pesquisa, juntamente com outras sugestões e ponderações necessárias.

## 2. REVISÃO DA LITERATURA

Para uma maior compreensão sobre o tema abordado, será realizado, no decorrer deste estudo, um detalhamento dos principais aspectos que estarão envolvidos com Link Open Data e com aplicações para a web. Sendo assim, este detalhamento levará em consideração alguns conceitos como: dados interligados, Resource Description Framework (RDF), web semântica, Sparql, Extensible Markup Language (XML) e ontologia, os quais estão diretamente ligados ao termo dados abertos interligados (Link Open Data). Por outro lado, linguagem de programação e a usabilidade referenciam tecnologias diretamente associadas à construção do software web.

### 2.1. Web Semântica

A web semântica é extensão da web atual que tem a intenção de estruturar o conteúdo da página web para que não só as pessoas o entendam, mas também as máquinas, de forma que se possa tornar a busca pela informação na web mais eficiente, automática e objetiva. Segundo Dziekaniak e Kirinus (2004, p. 21):

A Web Semântica visa incorporar semântica às informações. Isso proporcionará não somente aos usuários entenderem as informações como também as máquinas. Ela pretende fornecer estruturas e dar significado semântico ao conteúdo das páginas web, criando um ambiente onde agentes de software e usuários possam trabalhar de forma cooperativa.

O ambiente que a Web semântica está criando trará os dados estruturados de forma que a máquina consiga processar e entender, por meio de um conjunto de regras que ajudem a máquina a relacionar as palavras para busca e uma linguagem padrão, a qual todos usuários possam seguir e entender, conforme dito por Laufer (2015, p. 33).

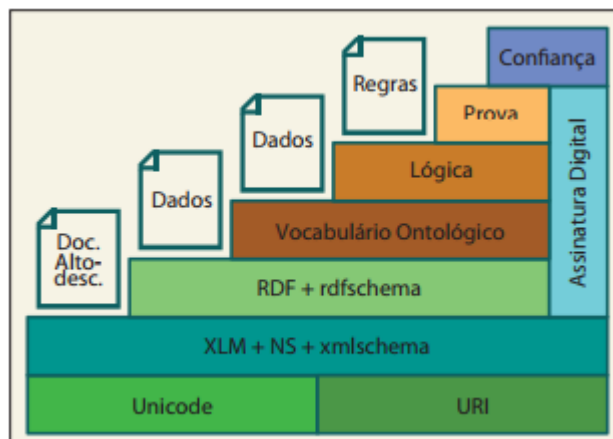
A Web Semântica busca facilitar o processo de comunicação entre os diversos participantes do ecossistema, de forma a criar um modelo mental comum, minimizando a possibilidade de ambiguidades, e facilitando, assim, o trabalho necessário para o desenvolvimento de aplicações que manipulem as diversas fontes de dados.

Com o avanço da Web Semântica, os softwares conseguiram executar tarefas mais complexas, automatizar tarefas simples, trazer informações mais precisas e

elaboradas em seus resultados, evitando o desperdício de tempo e a utilização dos recursos da máquina e dos dados de forma mais eficaz. Um exemplo disso seria a busca de uma casa para comprar, onde o sistema iria pesquisar a casa que melhor se encaixa na definição do usuário, mostrando a informações da localidade, histórico e as imobiliárias que possuem a casa para vender.

Para a implantação da Web Semântica, Tim Berners-Lee propôs uma arquitetura em camadas, onde cada camada possui uma função. Segundo Ramalho, Vidotti e Fujita (2005, p.5), o “modelo de arquitetura visa padronizar cada nível da construção da Web Semântica, mostrando como cada uma das tecnologias utilizadas relacionam-se e apresentando uma estrutura ideal para seu funcionamento”. A figura 1 mostra todas as camadas da arquitetura do modelo.

Figura 1 - Arquitetura da Web Semântica



Fonte: Dados Abertos Conectados

A Web Semântica possui uma arquitetura divididas em diferentes camadas, que interagem entre si, ou seja, possui um alto nível de interoperabilidade. Para compreendermos cada camada desta arquitetura e suas tecnologias, seguem, abaixo, suas definições:

Camada URL e Unicode: esta camada define quais símbolos podem representar os dados e como eles podem ser referenciados;

Camada XML, NS e XMLSchema: a camada que estabelece as regras básicas para uso da linguagem e de como será a estrutura da Web Semântica;

Camada RDF e RDFSchema: esta camada concebe um modelo de descrição da lógica de dados e um vocabulário a ser seguido.

Camada da Ontologia: a camada que define a relação entre as palavras e as conceituações entre elas.

Camada Lógica: é responsável pela definição das regras que tratam as informações.

Camada de Prova e Confiança: é a camada que cuida da parte dos testes, verificações e validações dos dados.

A Web Semântica define como tudo é construído e como são realizadas cada uma das atividades para o tratamento da informação. Os dados conectados são apenas uma parte desta arquitetura.

## **2.2. Dados interligados**

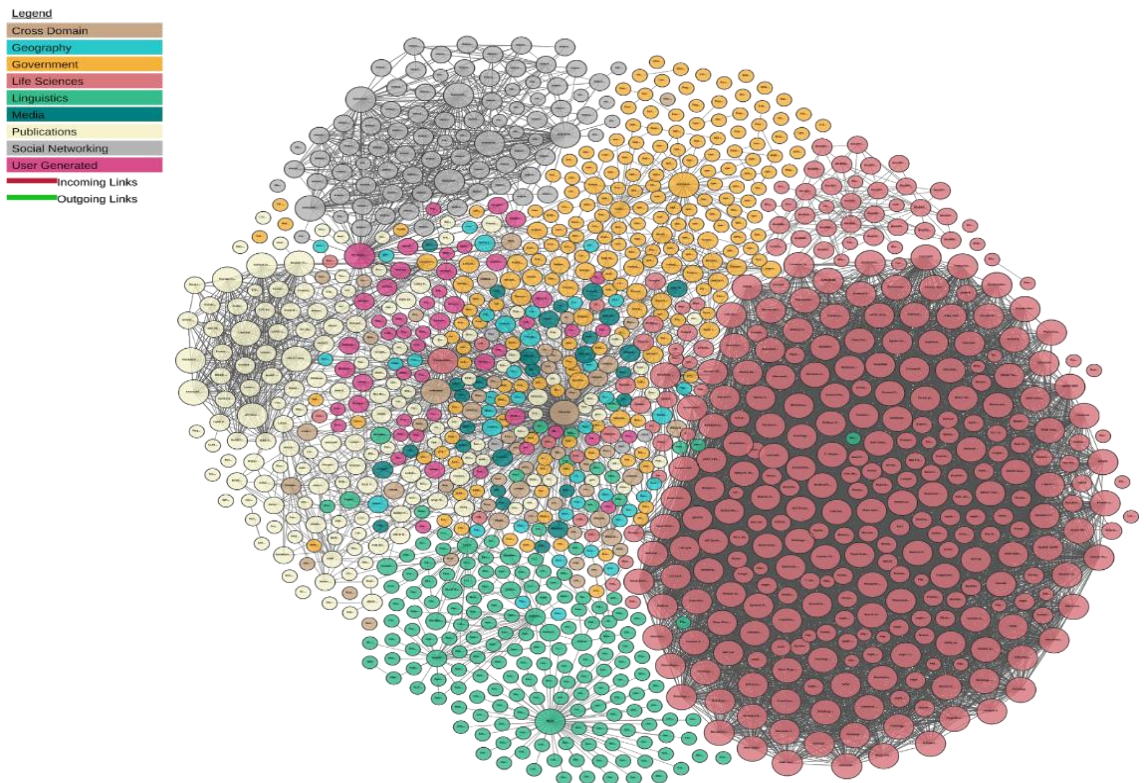
O conceito de dados interligados é referente ao conjunto de práticas para publicação de dados na web, que consiste em organizar os dados de forma sistemática, estruturada e interligando os dados entre si.

As aplicações de dados interligados abertos permitem a difusão do conhecimento a partir de informações de diversos setores da sociedade com a praticidade de realizar as interações entre as informações para o receptor.

Segundo Bizer, Heath e Berners-Lee (2009, p. 2), Linked Data refere-se a dados publicados na Web de tal forma que é legível por máquina, o seu significado é explicitamente definido, está ligado a outros conjuntos de dados externos e pode ser acessado a partir de conjuntos de dados externos.

O uso de dados interligados tem crescido muito durante os últimos anos, tornando-se uma grande nuvem de informações, possuindo diversos datasets interligados de diferentes segmentos como mídia, geografia, rede sociais, linguística, ciências biológicas, governo, entre outros. A figura 2 demonstra a nuvem de dados abertos conectados, apresentando as interligações de cada datasets, bem como a divisão de cada segmento. Mostra, também, estes mesmos datasets representados por uma cor diferente e quais deles pertencem a cada segmento. Ainda, na mesma figura, vemos como os datasets se interligam. No centro da grande nuvem está a DBpedia, que é a origem das primeiras ligações de primeiro nível entre os datasets.

Figura 2 – Nuvem de datasets Link Data



Fonte: LOD cloud diagram (2017)

Os dados conectados possuem alguns princípios para sua construção e publicação como mencionado por BIZER, HEATH e BERNES-LEE (2009): (1) uso de URIs como nomes para entidades; (2) uso de URIs via http, de modo que se possam buscar informações por esses nomes na web; (3) informações úteis associadas às URIs, usando padrões tais como RDF e SPARQL (W3C, 2008); (4) inclusão de associações com outras URIs, de modo que se possam descobrir mais entidades (Apud CAMPOS et al., 2013, p. 5).

Na Web, a utilização dos dados conectados é apenas uma parte da implantação ou transformação da web atual para a web semântica, de forma a tentar facilitar o acesso das máquinas às informações, assim como já é realizado pelas pessoas na web atual. Através deste acesso, facilita-se a captura dos dados e o processamento dos mesmos pelas máquinas.

Essa integração entre o homem e as máquinas irá deixar o desenvolvimento da web mais eficiente, podendo, assim, criar aplicações que possam beneficiar as pessoas,

como, por exemplo, a criação de aplicações para fiscalização de obras públicas. Nesta situação, a aplicação poderia demonstrar aos cidadãos informações como tempo de duração da obra, o gasto envolvido na obra, a empresa que está realizando a obra, o nível do progresso da obra e outras informações. Tudo isto por meio do acesso às bases de dados publicadas pelo governo. No entanto, esses dados publicados não trariam apenas vantagens para os cidadãos, mas, também, para empresas, que poderiam trabalhar com esses dados conectados com diversas bases do governo, ou de outras instituições, a fim de auxiliar seu trabalho e funcionamento. Especificamente, em relação a mudanças de legislação ou outros dados que interfeririam no funcionamento direto ou nos negócios da empresa.

De acordo com os autores Isotani e Bittencourt (2015, p. 34), a publicação dos dados conectados não necessariamente precisa ser em formato aberto. Como exemplo, uma entidade privada pode conectar dados, mas não necessariamente deixá-los abertos, como é o caso da Open Corporates. Há um grande incentivo para a criação de dados conectados de forma aberta, principalmente vindo de organizações como W3C, e também de um dos criadores da web Semântica: Tim Berners-Lee.

Tim Berners-Lee sugeriu um esquema de implantação das 5 estrelas para publicação dos dados conectados de forma aberta, conforme mostrado pela figura 3.

Figura 3 - Etapas do sistema de 5 estrelas



Fonte: 5stardata (2012)

1° estrela - É disponibilização dos dados na internet, por meio de uma licença aberta. Por exemplo: PDF

2° estrela - Tornar os dados disponíveis de forma estruturada, como, por exemplo, o que se usa em planilhas do Excel.

3° estrela - A disponibilização dos dados estruturados utilizando formatos não proprietários, como CSV.

4° estrela - Utilizar URI para identificar recursos de forma a facilitar o direcionamento dos recursos.

5° estrela - Conectar os seus dados com dados de outras pessoas para prover contexto.

A cada nível que se ultrapassa no sistema de 5 estrelas, maiores serão os benefícios do publicador e do consumidor. Além disso, esta passagem exige de ambos uma maior integração com o assunto tratado, já que, a cada migração de nível, os participantes se envolvem de forma mais profunda com o objeto de conhecimento, conforme o quadro 1 demonstra.

Como pode ser visto neste tópico, os dados conectados podem fornecer uma grande nuvem de informações, onde diversas pessoas e organizações podem extrair os dados conforme os seus interesses ou publicar mais dados para aumentar esta grande nuvem. As tendências são que o número de dados aumente com o passar do tempo e que tecnologias envolvidas neste processo sejam aprimoradas para que as máquinas consigam processar os dados mais eficientemente e que a quantidade dessas informações seja melhor aproveitada.

Quadro 1 - Sistema de 5 estrelas custo e benefícios

| Níveis     | Consumidor  | Publicador  |
|------------|---|---|
| ★          | <ul style="list-style-type: none"> <li>• Visualizar dados.</li> <li>• Imprimir.</li> <li>• Armazenar (no seu disco rígido, em um pendrive, ou na nuvem).</li> <li>• Introduzir os dados em qualquer outro sistema.</li> <li>• Alterar os dados sempre que você quiser.</li> <li>• Compartilhar os dados com quem desejar.</li> </ul>  | <ul style="list-style-type: none"> <li>• Simples para publicar.</li> </ul>  |
| ★★         | <ul style="list-style-type: none"> <li>• Pode processá-lo diretamente com um software proprietário para agregar, realizar cálculos, visualizações etc.</li> <li>• Exportar dados para outro formato (estruturado).</li> </ul>   | <ul style="list-style-type: none"> <li>• Simples para publicar.</li> </ul>  |
| ★★★        | <ul style="list-style-type: none"> <li>• Manipular os dados de qualquer forma sem as restrições de qualquer software específico.</li> </ul>   | <ul style="list-style-type: none"> <li>• Necessita de conversores ou plugins para exportar os dados do formato proprietário.</li> <li>• Simples para publicar.</li> </ul>   |
| ★★★★       | <ul style="list-style-type: none"> <li>• Apontar para ele de qualquer outro lugar (na Web ou localmente).</li> <li>• Marcar como favorito.</li> <li>• Reusar parte ou todos os dados.</li> <li>• Reutilizar ferramentas e bibliotecas existentes, mesmo se elas só entenderem partes dos padrões que o publicador utilizou.</li> <li>• Entender a estrutura de um "Grafo" de dados RDF pode ser mais útil do que dados tabulares (Excel/CSV) ou em árvore (XML/JSON).</li> <li>• Combinar os dados com outros dados seguramente. URIs são um esquema global, se duas coisas têm a mesma URI é porque é intencional. Assim, de certa forma, já está no caminho para ser dados 5 estrelas!</li> </ul> | <ul style="list-style-type: none"> <li>• Controle granular detalhado sobre os itens de dados e pode otimizar seus acessos (balanceamento de carga, cache, etc.)</li> <li>• Outros publicadores de dados podem referenciar seus dados, promovendo eles a 5 estrelas!</li> <li>• Normalmente investe algum tempo separando e fatiando seus dados.</li> <li>• Definir URIs para os itens de dados e pensar sobre como representar os dados.</li> <li>• Encontrar padrões existentes para reutilizar estes dados ou criar os seus próprios</li> </ul> |
| ★★★★★<br>★ | <ul style="list-style-type: none"> <li>• Descobrir mais dados (relacionados) enquanto consome os dados.</li> <li>• Diretamente aprender sobre o esquema dos dados.</li> <li>• Agora você precisa lidar com links de dados quebrados, assim como erro 404 nas páginas web.</li> <li>• Apresentar dados linkados como um fato é um risco porque permite que outras pessoas incluam conteúdo de qualquer página web nas suas páginas. Portanto, confiança e senso comum são necessários.</li> </ul>  | <ul style="list-style-type: none"> <li>• Torna os dados encontráveis.</li> <li>• Aumentar o valor dos seus dados.</li> <li>• Organização ganha os mesmos benefícios que os consumidores de dados já tem.</li> <li>• Investimento em recursos para ligar seus dados a dados de outros na Web.</li> <li>• Reparar links quebrados ou incorretos.</li> </ul>   |

Fonte: 5stardata

### 2.3. RDF



A Segundo os autores Campos et al. (2013, p. 5), “o RDF é um formato padrão para representação de dados na web. Esse formato permite que se representem fatos através de triplas na forma de sujeito, predicado e objeto que, por sua vez, representam entidades concretas ou abstratas do mundo real”. “O RDF é baseado na ideia de que os recursos são descritos através de declarações e possuem propriedades que têm valores” (LIMA e CARVALHO, 2005, p. 1).

O RDF é a estrutura que define como ficarão padronizadas as informações, permite que diversas aplicações possam lê-las sem a perda do significado e que pessoas possam visualizá-las. A base de dados RDF é feita de conjunto triplas, onde as triplas possuem a seguinte estrutura: Sujeito + Predicado + Objeto, Onde:

**Sujeito:** Identifica o Objeto.

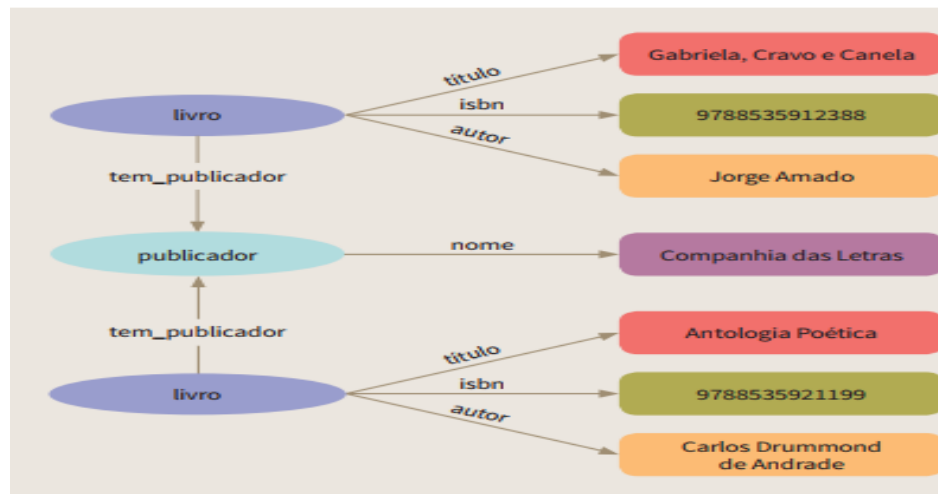
**Predicado:** É parte em que fica a propriedade ou a característica do sujeito.

**Objeto:** A parte em que fica o valor da propriedade.

De acordo com Schiessl (2015, p. 56), o RDF fornece uma descrição dos recursos utilizados em um esquema simples para expressar fatos ou declarações.

O RDF pode ser representado de diversas formas, sendo, uma delas, na forma de grafo, conforme figura 4, onde os sujeitos estão representados pelas elipses e os objetos são demonstrados pelas caixas. As linhas que interligam sujeito e objeto são os predicados, que podem descrever as URL utilizadas.

Figura 4 - Representação de grafos RDF



Fonte: Guia de web semântica, São Paulo (2015)

Outra forma, que é mais utilizada para a representação dos dados de RDF, é a utilização da sintaxe do XML, conforme figura 5. Nela, pode ser vista a linha 1, que faz a declaração dos prefix ou namespaces. Estes farão a ligação entre os sujeitos e os objetos e as demais linhas mostram as triplas RDF, aqui, exemplificando, uma busca realizada sobre “concept” na DBpedia.

Figura 5 - Representação RDF/XML - Informações DBpedia

```
<rdf:RDF xmlns:res="http://www.w3.org/2005/sparql-results#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<rdf:Description rdf:nodeID="rset">
<rdf:type rdf:resource="http://www.w3.org/2005/sparql-results#ResultSet" />
  <res:resultVariable>Concept</res:resultVariable>
  <res:solution rdf:nodeID="r0">
    <res:binding rdf:nodeID="r0c0"><res:variable>Concept</res:variable><res:value rdf:resource="
      http://www.w3.org/2002/07/owl#FunctionalProperty"/></res:binding>
  </res:solution>
  <res:solution rdf:nodeID="r1">
    <res:binding rdf:nodeID="r1c0"><res:variable>Concept</res:variable><res:value rdf:resource="
      http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/></res:binding>
  </res:solution>
</rdf:Description>
</rdf:RDF>
```

Fonte: DBpedia (2017)

Muitas dessas representações dos dados RDFs utilizam as referências URIs no lugar das palavras na declaração, pois a utilização das referências URI pode ajudar a

facilitar o desenvolvimento dos documentos e deixa as declarações com o formato já compartilhado na Web, conforme mencionado por Carvalho e Lima (2005, p. 7) abaixo:

O uso de referências URIs como sujeitos, predicados e objetos em declarações RDF permitem o desenvolvimento e uso de vocabulários compartilhados na Web, desde que pessoas possam descobrir e começar a usar vocabulários criados por outros, refletindo em um entendimento compartilhado destes conceitos. Este compartilhamento de vocabulários é um dos objetivos da Web Semântica.

## 2.4. SPARQL

O Sparql (Simple Protocol and RDF Query Language) é uma linguagem de consulta para os grafos de RDF, como se fosse o SQL para as bases de dados convencionais. O Sparql permite que o usuário faça as buscas nas Triplas RDF para ter o retorno de algum valor. De acordo com Berner-Lee (W3C,2008), “Tentar usar a web semântica sem SPARQL é como tentar usar um banco de dados relacional sem SQL”. (Apud CONEGLIAN e SEGUNDO, 2016, p. 90).

O Sparql realiza as consultas para a extração de informações da base de dados de RDF. Ele possui vocábulos e sintaxe específicos para manipulação e tratamento da informação das triplas RDF e permite que o usuário faça busca complexa utilizando filtros nas consultas, fazendo com que só sejam extraídas as informações necessárias ou específicas da base, assim como descrito por Schiessl:

Ela permite extrair dados (semi) estruturados, explorar dados utilizando consultas por relações desconhecidas, executa combinações complexas de bases de dados heterogêneas com consultas simples, transforma dados RDF de um vocabulário para outro e constrói novos grafos RDF a partir de consultas em outros vocabulários. (2015, p. 68)

A sintaxe do SPARQL, apesar de ter alguns comandos parecidos com os do SQL, tem comandos e estrutura própria para o tratamento das informações, conforme mostrado na figura 6:

Figura 6 - Consulta SPARQL

```
select distinct ?Concept where {
  [] a ?Concept
} LIMIT 100|
```

Fonte: DBpedia (2017)

**PREFIX** - Utilizado para fornecer uma abreviação para namespaces. Logo, ele faz uma abreviação das URI para simplificar código.

**SELECT** - Utilizado para realizar as consultas na base RDF retirando as informações da base e apresentando em forma de tabela.

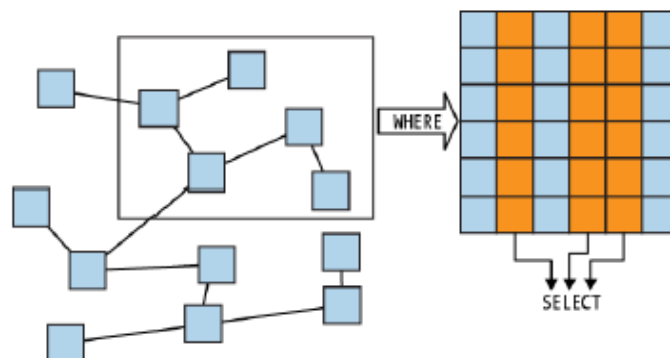
**CONSTRUCT** - Comando utilizado para transformar o resultado da pesquisa em uma tripla RDF.

**DISTINCT** - Remover linhas duplicadas.

As variáveis - As variáveis são iniciadas por “?” e podem representar qualquer posição da tripla RDF.

**WHERE** - Coloca as restrições na consulta, ou seja, só mostrará os resultados que atendam as condições colocadas pelo usuário.

Figura 7 - Utilização do Where e Select



Fonte: Learning SPARQL: Querying and Updating with SPARQL 1.1 (2011)

A maneira como o SPARQL é utilizado pelos usuários pode trazer resultados bastante detalhados, estruturados e de modo otimizado, mesmo utilizando os conceitos básicos da linguagem. Ainda há muitas informações a serem aprendidas no SPARQL, sendo que muitas dessas informações podem ser encontradas no próprio site W3C.

## 2.5. Ontologia

A ontologia é definida pelo manual dos dados abertos e desenvolvedores web como:

Na filosofia, ontologia é o estudo da existência ou do ser enquanto ser, ou seja, a maneira de compreender as identidades e grupos de identidades. Na ciência da computação, é um modelo de dados que representa um conjunto de conceitos sob um domínio e seus relacionamentos, ou, mais formalmente, especifica uma conceitualização dele. (2011, p. 19)

A ontologia é um fator importante para a web semântica, pois ele deixa definido de forma clara os conceitos e os relacionamentos entre eles, podendo ser vista como um modelo do conhecimento. A definição de ontologia, na concepção de Breitman, é dada abaixo:

Ontologias são especificações formais e explícitas de conceitualizações compartilhadas. Ontologias são modelos conceituais que capturam e explicitam o vocabulário utilizado nas aplicações semânticas. Servem como base para garantir uma comunicação livre de ambigüidades. Ontologias serão a língua franca da Web Semântica. (BREITMAN, 2005, p. 7)

Para utilização da ontologia, são seguidos alguns princípios que devem ser a clareza, coerência, extensibilidade, compromissos ontológicos e de codificação mínimos. Alguns benefícios que a ontologia pode fornecer para os sistemas baseados em conhecimentos são: o compartilhamento de informações e as definições dos conceitos, de maneira que os termos tenham clareza e coerência; a reutilização dos conceitos já existentes para a criação de novos conceitos ou para adaptação dos conceitos já existentes, permitindo que o conceito e a semântica sejam entendidos pelas máquinas, propiciando a interoperabilidade do sistema.

Segundo Morais e Ambrósio (2007, p. 6) a ontologia pode ser classificada em cinco categorias:

**Ontologias Genéricas:** São consideradas ontologias “gerais”. Descrevem conceitos mais amplos, como elementos da natureza, espaço, tempo, coisas, estados, eventos, processos ou ações, independente de um problema específico ou domínio particular.

**Ontologias de Domínio:** Descrevem conceitos e vocabulários relacionados a domínios particulares, tais como medicina ou computação, por exemplo.

**Ontologias de Tarefas:** Descrevem tarefas ou atividades genéricas, que podem contribuir na resolução de problemas, independente do domínio que ocorrem. Por exemplo, processos de vendas ou diagnóstico.

**Ontologias de Aplicação:** Descrevem conceitos que dependem tanto de um domínio particular quanto de uma tarefa específica.

**Ontologias de Representação:** Explicam as conceituações que fundamentam os formalismos de representação de conhecimento, procurando tornar claros os compromissos ontológicos embutidos nestes formalismos.

Com isso, a ontologia ocupa um papel importante para a construção dos sistemas baseados em informação, pois facilita o desenvolvimento do sistema com o reuso destas informações, permite a interoperabilidade entre os sistemas e melhora a eficácia nas buscas semânticas. Isso tudo sendo possível devido à clareza com que a ontologia define seus conceitos.

## 2.6. OWL

A Web Ontology Language (OWL) é uma linguagem que visa a descrição das ontologias Web, permitindo ao usuário analisar e definir a estrutura de classes da ontologia, com intuito de entender a lógica e as relações destas classes e, assim, facilitar a busca de inconsistências ou falhas, cuja vinculação à ontologia é dada pela semântica.

O OWL utiliza o vocabulário do RDF para a definição da estrutura-base dos dados, mas a forma como serão vistas as relações dos conceitos vem do próprio OWL, assim como mencionado por Santos e Carvalho:

Uma ontologia OWL pode incluir: relações de taxonomia entre classes; propriedades dos tipos de dados e descrições dos atributos de elementos das classes, propriedades do objeto e descrições das relações entre elementos das classes, instâncias das classes e instâncias das propriedades. (2013, p. 8)

O OWL possui três versões para serem utilizadas. O que as diferencia é a expressividade da linguagem, ou seja, seu diferencial é na quantidade de vocábulos e no nível de complexidade da organização. As versões do OWL são OWL Full, OWL DL e OWL Lite, conforme mencionado por Lopes (2017, p. 34):

**OWL Full:** A versão mais expressiva do OWL. Permite o uso de todas as funcionalidades disponíveis. Porém, tamanha expressividade compromete a eficácia na extração e inferência de informações bem como a decidibilidade da ontologia. Uma ontologia decidível é aquela capaz de ser processada em tempo hábil por um reasoner;

**OWL DL (Description Logic):** Subconjunto da versão Full que tem como objetivo prover certo nível de expressividade sem comprometer a eficiência de tarefas como a inferência (reasoning) de informações por parte dos motores de inferência (reasoner);

**OWL Lite:** A menos expressiva das versões de OWL. Tem sido muito usada para definição de domínios simples. Pode ser uma boa alternativa para migração de tesouros. A principal vantagem é a facilidade de entender e de se implementar uma ontologia usando esta versão. A desvantagem é a restrição na sua expressividade.

As versões do OWL são diferenciadas principalmente pelo nível de expressividade contidas nelas, pois quanto maior o nível de expressividade da versão maior será o nível de complexidade da ontologia, de sua descrição e atividades atribuídas, sendo que o OWL Full é mais expressivo. Por essa razão, toda ontologia criada no OWL Lite e OWL DL são válidas para OWL Full e toda ontologia feita no OWL Lite é válida no OWL DL.

O OWL possui alguns componentes para a estruturação da ontologia de todas as relações entre os termos envolvidos na ontologia.

**Individuals** - Indivíduos representam objetos no domínio de interesse (ou domínio do discurso).

**Properties** - Propriedades são relações binárias (relações que contém duas coisas) entre indivíduos, ou seja, as propriedades ligam dois indivíduos.

**Classes** - As classes OWL são conjuntos que contêm os indivíduos. Elas são descritas formalmente (descrições matemáticas) de forma que sejam apresentados os requisitos para a participação na classe.

## 2.7. Análise de Requisito Não Funcional

Os requisitos não funcionais são aqueles que não se referem diretamente à necessidade do software e suas funcionalidades. Podemos citar, como exemplo, aspectos relacionados ao desempenho, a confiabilidade, segurança, disponibilidade, tamanho, usabilidade, robustez e muitos outros, assim como definido por Sommerville (2011, p. 60): “Os requisitos não funcionais, como o nome sugere, são requisitos que não estão diretamente relacionados com os serviços específicos oferecidos pelo sistema a seus usuários”. Algumas propriedades definidas na visão de Bezerra (2007, p. 23):

**Confiabilidade:** corresponde a medidas quantitativas da confiabilidade do sistema, tais como tempo médio entre falhas, recuperação de falhas ou quantidade de erros por milhares de linhas de código-fonte.

**Desempenho:** requisitos que definem tempo de resposta esperados para as funcionalidades do sistema.

**Portabilidade:** restrições sobre as plataformas de hardware e de software nas quais o sistema será implantado e sobre o grau de facilidade para transportar o sistema para outras plataformas.

**Segurança:** limitações sobre a segurança do sistema em relação a acessos não-autorizados.

**Usabilidade:** requisitos que se relacionam ou afetam a usabilidade do sistema. Exemplos incluem requisitos sobre a facilidade de uso e a necessidade ou não de treinamento dos usuários.

Os requisitos não funcionais muitas vezes são fatores tão críticos quanto os requisitos funcionais, pois podem causar o mau funcionamento de todo o sistema e não somente de uma parte específica. Por essa razão, devem ser identificados e verificados junto ao cliente, já que alguns destes são trazidos por exigência do próprio cliente, como demonstrado por Sommerville (2011, p. 61):

Os requisitos não funcionais surgem por meio das necessidades dos usuários, devido a restrições de orçamento, políticas organizacionais, necessidade de interoperabilidade com outros sistemas de software ou hardware, ou a partir de fatores externos, como regulamentos de segurança ou legislações de privacidade.

Muitos requisitos não funcionais tornam-se complexos quanto a sua definição e mensuração, já que são ditados, muitas vezes, por um cliente que não possui experiência ou conhecimento do sistema que está sendo criado, levando a informações incorretas ou inserindo propriedades que causarão danos a aplicação.

## 2.8. Análise de Requisito Funcional

Os requisitos funcionais são o que definem as funcionalidades do sistema, ou seja, descrevem as funções do software de forma a detalhar como será realizada a atividade pelo software. Descrição dos requisitos funcionais pelo autor Sommerville:



Os requisitos funcionais de um sistema descrevem o que ele deve fazer. Eles dependem do tipo de software a ser desenvolvido, de quem são seus possíveis usuários e da abordagem geral adotada pela organização ao escrever os requisitos (2011, p. 59).

Esta etapa de definição é importante para a construção do software, pois nesta fase é que os desejos e necessidades do cliente começaram a ser elaborados e a tomarem forma. Desta maneira, qualquer erro ou alinhamento incorreto realizado com o cliente e sua elaboração e definição de requisitos poderá causar a insatisfação do mesmo e o retrabalho para as equipes de desenvolvimento. Por isso, validações e realinhamentos junto ao cliente devem ser constantes, de forma que os requisitos funcionais estejam muito bem avaliados, organizados e documentados, para não acabar gerando prejuízos, retrabalho e resultado insatisfatório.

Segundo Sommerville (2011, p. 60):

A imprecisão na especificação de requisitos é a causa de muitos problemas da engenharia de software. É compreensível que um desenvolvedor de sistemas intérprete um requisito ambíguo de uma maneira que simplifique sua implementação. Muitas vezes, porém, essa não é a preferência do cliente, sendo necessário, então, estabelecer novos requisitos e fazer alterações no sistema. Naturalmente, esse procedimento gera atrasos de entrega e aumenta os custos.

## **2.9. Linguagem de Programação**

A linguagem de programação é o método para passar instruções para o computador através do programa de computador. Desta forma, o computador manipula os dados ou valores de acordo com intenção de quem está programando.

Há diversas linguagens de programação. Para se criar um software, emprega-se determinada linguagem, cada uma com sua peculiaridade, ou seja, cada linguagem possui uma sintaxe diferente, uma forma diferente de execução e de estruturação, uma utilização maior em ramos diferentes e até mesmo um tempo diferenciado na execução de ações. Cada vez mais surgem novas linguagens no mercado visando à construção de programas que possam atender as nossas necessidades, ou facilitar a vida das pessoas.

Existem várias maneiras de ser classificada as linguagens de programação, como por exemplo:

**Compilada:** Onde é realizada uma tradução do código do programa para a linguagem de máquina criando um executável ou biblioteca. A tradução é feita pelo Compilador;

**Interpretada:** conforme é traduzida a linguagem já é realizado a execução. O responsável por realizar a tradução é o Interpretador;

**Linguagem de alto nível:** Uma linguagem mais próxima do entendimento humano onde o seu nível de abstração é muito maior, se distanciando da linguagem de máquina;

**Linguagem de baixo nível:** É mais próximo da linguagem de máquina. Sua construção está diretamente ligada ao conhecimento da arquitetura do computador e instruções do processador.

Outro aspecto utilizado para distinguir as linguagens é referente ao estilo de programação como mostrado na Wikipédia:

**Programação estruturada:** é uma forma de programação de computadores que preconiza que todos os programas possíveis podem ser reduzidos a apenas três estruturas: sequência, decisão e repetição;

**Programação modular:** é uma forma de programação na qual o desenvolvimento das rotinas de programação é feito através de módulos, que são interligados entre si através de uma interface comum;

**Programação Orientada a Objetos (POO):** é um paradigma de análise, projeto e programação de sistemas de software baseado na composição e interação entre diversas unidades de software chamadas de objetos.

## 2.10. Usabilidade

A usabilidade está ligada à facilidade que usuário tem para usar sua interface. A ISO 9241 define usabilidade como a capacidade de um produto ser utilizado por usuários específicos para atingir objetivos específicos com eficácia, eficiência e satisfação em um contexto específico de uso.

Para Nielson (2012), a usabilidade tem 5 componentes de qualidade:

**Aprendizagem:** A interface deve ser simples o suficiente para que usuário consiga aprender de forma rápida a utilizá-la.

**Eficiência:** Depois que o usuário aprende a utilizar a interface, ele deve conseguir realizar as tarefas de forma rápida.

**Memorabilidade:** Mesmo que usuário fique algum tempo sem utilizar a interface, deve ser simples o suficiente para que o usuário lembre rapidamente como utilizá-la.

**Erros:** Mensagens de erros devem ser amigáveis, a interface deve possuir o mínimo de erros possíveis e ser facilmente compreendida pelo usuário para não cometer erros.

**Satisfação:** Quão satisfatória é sua interface para usuário?

## 2.11. XML

O XML é uma metalinguagem de marcação. Seu uso é recomendado pela W3C, pois possui flexibilidade alta podendo criar diversos tipos de tags e serem usadas em diferentes plataformas. Possui uma estrutura simples feita para ser entendida tanto por máquinas como para pessoas e sua organização é de forma hierárquica.

O XML permite uma grande liberdade para a construção de diversos documentos, pois, diferente de outras linguagens, ele permite que o autor defina as marcas ou tags do documento, podendo, assim, montar diversos tipos de documentos com objetivos diferentes, como referenciado por Almeida (2002, p. 6):

O XML é uma arquitetura que não possui elementos\* e marcas predefinidas. Não especifica como os autores vão utilizar metadados, sendo que existe total liberdade para utilizar qualquer método disponível, desde simples atributos\*\*, até a implementação de padrões mais complexos.

Outra característica bastante vantajosa do XML é sua portabilidade, que vem devido sua grande flexibilidade, pois uma aplicação pode gerar um código XML para que outra aplicação possa ler e usar essas informações, como os bancos de dados.

Figura 8 - Código XML

```

<?xml version="1.0" encoding="utf-8" ?>
<Times>
  <Time Nome="Nome 1" Cidade="Cidade 1">
    <!--Atributos do time 1-->
    <Vitorias>Vitorias 1</Vitorias>
    <Empates>Empates 1</Empates>
    <Derrotas>Derrotas 1</Derrotas>
  </Time>
  <Time Nome="Nome 2" Cidade="Cidade 2">
    <!--Atributos do time 2-->
    <Vitorias>Vitorias 2</Vitorias>
    <Empates>Empates 2</Empates>
    <Derrotas>Derrotas 2</Derrotas>
  </Time>
  <Time Nome="Nome 3" Cidade="Cidade 3">
    <!--Atributos do time 3-->
    <Vitorias>Vitorias 3</Vitorias>
    <Empates>Empates 3</Empates>
    <Derrotas>Derrotas 3</Derrotas>
  </Time>
</Times>

```

Fonte: Macoratti (2008)

Para a construção de um documento XML, como na figura 8, precisamos entender algumas estruturas básicas que são os elementos (estrutura representada por texto entre os sinais '<' e '>'), os atributos (característica dos elementos), comentários (demarcações nos documentos que não serão processadas pelo programa) e as instruções de processamento (estrutura responsável pela forma de interpretação do documento).

## 2.12. HTML

O HTML (Hypertext Markup Language) é uma linguagem de marcação utilizada para as construções dos sites pelos navegadores web. Assim como citado por Eris e Ferreira (2012, p. 27), o HTML é uma linguagem para publicação de conteúdo (texto, imagem, vídeo, áudio e etc...) na Web.

O HTML é uma linguagem criada na década de 90 e, com passar do tempo, ela foi sendo atualizada e ganhando novas versões, como o HTML 2.0, HTML 3.2, HTML 4.01, XHTML e o mais atual, o HTML5, que veio com diversas mudanças em relação às versões anteriores, colocando novas funcionalidade e ferramentas, como integração de conteúdo

multimídia sem a necessidade de plug-ins. Os autores Eris e Ferreira (2012, p. 32) citam que:

O HTML5 modifica a forma de como escrevemos código e organizamos a informação na página. Seria mais semântica com menos código. Seria mais interatividade sem a necessidade de instalação de plugins e perda de performance. É a criação de código interoperável, pronto para futuros dispositivos e que facilita a reutilização da informação de diversas formas.

A sintaxe do HTML é formada por marcações ou tags que irão estruturar a página web e dar sentido ao conteúdo inserido nela, podendo definir seus papéis, atributos e hierarquias. Sendo assim, as tags transformam os elementos inseridos pelo desenvolvedor e dão a eles um significado na página web. Também, elas atribuem características a esses elementos e uma ordem ou hierarquia na página. Isto pode ser observado na figura 9, onde a tag <img> é utilizada para inserir imagem dentro da página e, nesta imagem, também colocamos as características desta imagem, através do atributo style, como a altura. Também, mostramos a hierarquia da tag <img>, através da relação da tag <a> ou da própria tag <body>.

Figura 9 - Tags HTML

```
<!DOCTYPE html>
<html>
<body>

<p>The image is a link. You can click on it.</p>

<a href="default.asp">
  
</a>

<p>Add "border:0;" to prevent IE9 (and earlier) from displaying a border around
the image.</p>

</body>
</html>
```

Fonte: w3schools.com (2017)

## 2.13. PHP

O PHP (Personal Home Page) é uma linguagem utilizada para o desenvolvimento de aplicações web, sendo que sua parte presente fica no lado servidor,

ao qual os usuários não possuem, tornando impossível visualizar o código-fonte da aplicação. PHP é assim descrita pelo seu manual:

A melhor coisa em usar o PHP é que ele é extremamente simples para um iniciante, mas oferece muitos recursos para um programador profissional. Não tenha medo de ler a longa lista de funções do PHP. Pode entrar com tudo, o mais rápido que puder, e comece a escrever scripts simples em poucas horas. (MANUAL PHP, 2017)

A linguagem surgiu em 1995, com Rasmus Lerdorf, e já foram criadas diversas versões que foram sendo implementadas por novas funcionalidades, correções de bugs e de segurança, desembocando na versão mais atual, que é versão 7 do PHP. Com estes avanços, muitos detalhes dentro do código foram mudando, mas suas características básicas permanecem. Estas características fundamentais são mencionadas por Niederauer (2011):

**Gratuito e com código aberto:** O PHP possui o código-fonte aberto para todos as pessoas. Além de possuir uma documentação detalhada sobre o código-fonte, também pode ser baixado, nas suas versões mais recentes, no site oficial, de forma gratuita.

**Embutido no HTML:** A linguagem é embutida no HTML, ou seja, quando uma página tem a extensão .php, sempre que servidor for realizar as requisições, ele saberá que esta página contém linguagem de programação, assim, interpretando o HTML e o PHP, não importando a ordem.

**Baseado no servidor:** O código PHP é executado no lado servidor, enviando ao cliente apenas os resultados desta execução, fazendo com que a parte do código da aplicação não fique exposto ao usuário ou perceptível ao usuário. Exige menos processamento do computador do usuário devido o carregamento do código e a execução se dar toda no servidor.

**Bancos de dados:** O PHP possui suporte a diversos tipos de banco de dados, sendo capaz de realizar diversas funções de cada banco de dados em seu código. Entre estes, estão MySQL, PostgreSQL, Sybase, Oracle, SQL Server e muitos outros.

**Portabilidade:** A linguagem é multiplataforma, podendo ser executada em diferentes sistemas operacionais, como Linux, Unix, Windows, MAC OS e entre outros.

## 2.14. CSS

Na construção das páginas web são utilizados o HTML para a criação da estrutura da página e, para a estilização da página, utiliza-se o CSS (Cascading Style Sheets). A função do css fica na formatação dos elementos do site, ou seja, ele insere elementos ou atribui características para algumas tags do HTML, alterando seu aspecto visual, e tornando a interface gráfica mais agradável para os usuários.

Para Eris e Ferreira (2012, p. 23), o CSS pode ser descrito da seguinte maneira:

Formatará o conteúdo de forma que seja visualmente agradável em qualquer meio de acesso. A informação é acessada por diferentes meios de acesso, desde sistemas de busca até aparelhos como tablets, smartphone etc o CSS é o responsável por formatar a informação para que ela seja consumida em qualquer meio de acesso de forma simples.

Para a formatação com o CSS é preciso, antes, realizar a construção das regras de negócio, ou seja, verificar quais os elementos que serão alterados e como eles serão alterados, definindo, assim, todos seletores e blocos de declaração necessários para a modificação do site.

## **2.15. JavaScript**

O JavaScript é uma linguagem que surgiu em 1995, criada pelo desenvolvedor Brendan Eich da Netscape. Apesar do seu nome ser parecido com Java, essa linguagem não tem nenhuma ligação direta com a mesma. Nos dias atuais é uma linguagem bastante popular e utilizadas em sites devido ao dinamismo que a linguagem oferece a uma aplicação web.

A linguagem JavaScript é descrita e caracterizada por Flanagan (2013, p. 1) como uma linguagem de alto nível, dinâmica, interpretada e não tipada, conveniente para estilos de programação orientados a objetos e funcionais.

O JavaScript é uma linguagem do tipo client-side (Cliente-servidor). Sendo assim, é executada somente no computador do usuário, não necessitando do processamento do código pelo servidor. Ao contrário da linguagem como PHP, o JavaScript fica presente no código junto ao HTML e o CSS, mas acessível na máquina do usuário, visualizando-se o código-fonte da página web no navegador.

Conforme Flanagan (2013, p. 2), “A linguagem JavaScript básica define uma API mínima para trabalhar com texto, arrays, datas e expressões regulares, mas não inclui

funcionalidade alguma de entrada ou saída”, já que está funcionalidade é tarefa do navegador web.

## 2.16. Framework

Os Frameworks são como caixas de ferramentas que podem possuir diversas funcionalidades a serem implementadas dentro do código de software. Muitas vezes, para facilitar ou diminuir o trabalho do desenvolvedor, eles podem oferecer templates prontos, códigos de validação, operações simples, estruturação para programa, entre outras funcionalidades.

O framework é definido por Willemann e Ibarra como:

Um framework ou arcabouço é uma estrutura de suporte definida em que um outro projeto de software pode ser organizado e desenvolvido, quando se analisa o conceito no âmbito do desenvolvimento de software. Um framework pode incluir programas de suporte, bibliotecas de código, linguagens de script e outros softwares para ajudar a desenvolver e juntar diferentes componentes de um projeto de software. (2007, p. 41)

A utilização dos frameworks na construção de software pode trazer algumas vantagens como redução do custo, devido ao gasto menor do tempo do desenvolvedor na aplicação, facilidade na implantação de algumas funções, ajudando o desenvolvedor a manter foco nas partes essenciais do programa. Permite a reutilização do código e aumenta eficiência no desenvolvimento do software.

De acordo com o Willemann e Ibarra, para ser considerado um framework, ele precisa ter algumas características básicas:

- Precisa ser reutilizável.
- Precisa facilitar o desenvolvimento de sistemas.
- Precisa possuir boa documentação.
- Precisa ser completo para o que se propõe.
- Precisa ser eficiente.

O Framework, como já observado pelas características básicas, é construído para a facilitação do desenvolvimento do sistema e também com o intuito de auxiliar no desenvolvimento do software. Temos, como alguns exemplos de Framework da atualidade, o Laravel, Pure, Bootstrap, Symphony, etc.



## 2.17. Laravel

O Laravel é framework para desenvolvimento de sistemas web em PHP que utiliza o modelo padrão MVC (Model, View, Control) como estrutura dos projetos. O objetivo desse framework é facilitar o desenvolvimento do código de forma mais simples e limpa. Auxilia a melhora da criação de código relacionado a certas características do projeto, como, por exemplo, uso de banco de dados, direcionamento de rotas e templates.

O Laravel utiliza o Blade Engine para parte visual do projeto. Através dessa engine, ele consegue fazer controle dos templates da aplicação, permitindo que não se necessite ter repetição de código. Para tanto, ele se utiliza de herança e seções para oferecer o reaproveitamento deste código. Vemos isto na figura 10, onde as `@section` e `@extends` são templates já desenvolvidas, e que foram chamadas para construção da página, podendo ser incluídas em outras páginas sem a necessidade de ficar montando toda parte de layout novamente com HTML e CSS.

Figura 10 - Código utilizando heranças e seções

```
<!-- Stored in resources/views/child.blade.php -->

@extends('layouts.app')

@section('title', 'Page Title')

@section('sidebar')
    @parent

    <p>This is appended to the master sidebar.</p>
@endsection

@section('content')
    <p>This is my body content.</p>
@endsection
```

Fonte: Laravel (2017)

ORM Eloquent é a ferramenta implementada no Laravel, que facilita o controle do banco de dados nas aplicações, trazendo diversas funcionalidades relacionados às operações CRUD (Create, Read, Update, Delete) das aplicações. Além dessas facilidades, o Laravel possui simples e rápida configuração de conexão a banco de dados, tendo suporte a 4 tipos de banco de dados, a saber: Mysql, MySQL, PostgreSQL, SQLite e SQL Server.

A documentação do Laravel possui muito conteúdo e detalhes para explicar seu funcionamento e ferramentas inclusas, podendo, assim, proporcionar um aprendizado mais rápido e simples para seus usuários.

### 3. PROCEDIMENTOS METODOLÓGICOS

Para realização desta pesquisa, são oferecidas diversas técnicas e métodos. O emprego de um destes métodos ou técnicas em particular é que define o tipo de pesquisa a ser executada. Segundo Silva e Menezes (2005, p. 20), a pesquisa aqui desenvolvida pode ser classificada da seguinte maneira:

Do ponto de vista da natureza da pesquisa, ela é aplicada, pois “objetiva gerar conhecimentos para aplicação prática e dirigidos à solução de problemas específicos. Envolve verdades e interesses locais. (SILVA e MENEZES, 2005, p. 20)”

A forma como foi feita a abordagem da pesquisa é qualitativa, já que, conforme mencionado por Silva e Menezes (2005, p. 20), a pesquisa não precisa de métodos ou técnicas que envolvam números ou estatísticas. Ela tem o foco principal na descrição do processo. O processo e seu significado são os itens principais de abordagem.

Do ponto de vista de seus objetivos, ela se classifica como exploratória, devido ao fato de apresentar como a “principal finalidade desenvolver, esclarecer e modificar conceitos e ideias” (Gil, 2002, p. 27), sendo que a pesquisa é desenvolvida “com o objetivo de proporcionar visão geral, de tipo aproximativo, acerca de determinado fato.” (GIL, 2002, p. 27).

Do ponto de vista dos procedimentos técnicos, pode ser classificada em bibliográfica, já que “é desenvolvida com base em material já elaborado, constituído, principalmente, de livros e artigos científicos (GIL, 2002, p. 44)”, considerando-se sua base conceitual, e estudo de caso, pois “consiste no estudo profundo e exaustivo de um ou poucos objetos, de maneira que permita seu amplo e detalhado conhecimento (GIL, 2002, p. 54)”.

Para a realização do trabalho houve uma divisão, por etapas, de todos os passos necessários para a conclusão da pesquisa. Apesar de existirem diversos métodos e técnicas para a consecução da mesma, optou-se, na primeira fase, pela pesquisa bibliográfica, onde foram analisadas obras, artigos e documentos de autores que trabalharam com o tema, para dar suporte ao projeto.

A pesquisa bibliográfica foi desenvolvida com base em material já elaborado, constituído principalmente de livros e artigos científicos. De acordo com GIL (2002, p. 45), a principal vantagem da pesquisa bibliográfica reside no fato de permitir ao investigador a

cobertura de uma gama de fenômenos muito mais ampla do que aquela que poderia pesquisar diretamente.

A segunda etapa da pesquisa referiu-se à parte de análise de requisitos do protótipo, pois este processo é essencial para a construção e o melhor entendimento do software. Nesta etapa foi onde especificamos elementos como funcionalidade da aplicação, fluxo de atividade deste software e especificidade relacionada às tecnologias utilizadas.

A análise de requisitos foi classificada como parte fundamental para sucesso do desenvolvimento do software, pois documenta a estrutura do software, facilitando a busca de inconsistências, pontos de melhorias e visão do software, contribuindo, também, para fornecer critérios para a avaliação da qualidade do software.

A terceira etapa foi voltada a desenvolvimento das atividades e integrações exigidas pelo software, ou seja, a explicação e demonstração da parte do código, com imagens, das funcionalidades do mesmo, além de ser também etapa que mostra a maneira que o layout foi construído e as interações entre as telas, para um melhor entendimento do sistema.

A última etapa é a de teste, onde foram inseridos bugs no sistema para achar erros no código-fonte do protótipo, a fim de se aperfeiçoá-lo. Para a realização deste teste, foram utilizados framework especializados para depurar o código-fonte e encontrar os problemas, podendo ser feitos via teste simples - para achar erros de sintaxe - ou testes unitários e de integração.

## **4. DESENVOLVIMENTO**

### **4.1. Ferramentas utilizadas**

#### **4.1.1. Astah**

Ferramenta utilizada para a modelagem das UML. Criado por uma empresa no Japão chamada Change Vision, Astah é um software desenvolvido em Java e sua base de criação de UML também é focada em Java. No presente estudo, a aplicação foi utilizada para elaborar os diagramas de classe e os casos de uso. Apesar da aplicação não utilizar o PHP como a linguagem principal, os diagramas fornecidos por esta ferramenta podem ser utilizados em qualquer linguagem.

#### **4.1.2. Sublime Text**

Software multiplataforma para edição de texto desenvolvido na linguagem Python com o objetivo de ser simples e fácil de utilizar, possui diversos recursos e funcionalidades que podem ajudar no desenvolvimento da codificação.

#### **4.1.3. Pingendo**

É uma aplicação desenvolvida para a criação de páginas e sites responsivos muito simples e fácil de utilizar. Ela emprega o Bootstrap como base para criação das páginas HTML. Com essa aplicação, é possível criar sites apenas arrastando os elementos desejados, além de permitir visualizar o código que está sendo gerado pela aplicação.

### **4.2. Base de dados em RDF**

Para o desenvolvimento do protótipo, foram analisadas algumas bases de dados RDF que permitissem o teste da aplicação. Dentre as bases estudadas, tivemos o DBpedia, Wikidata e o SIOP. A opção mais adequada para o teste do protótipo a DBpedia, por ser

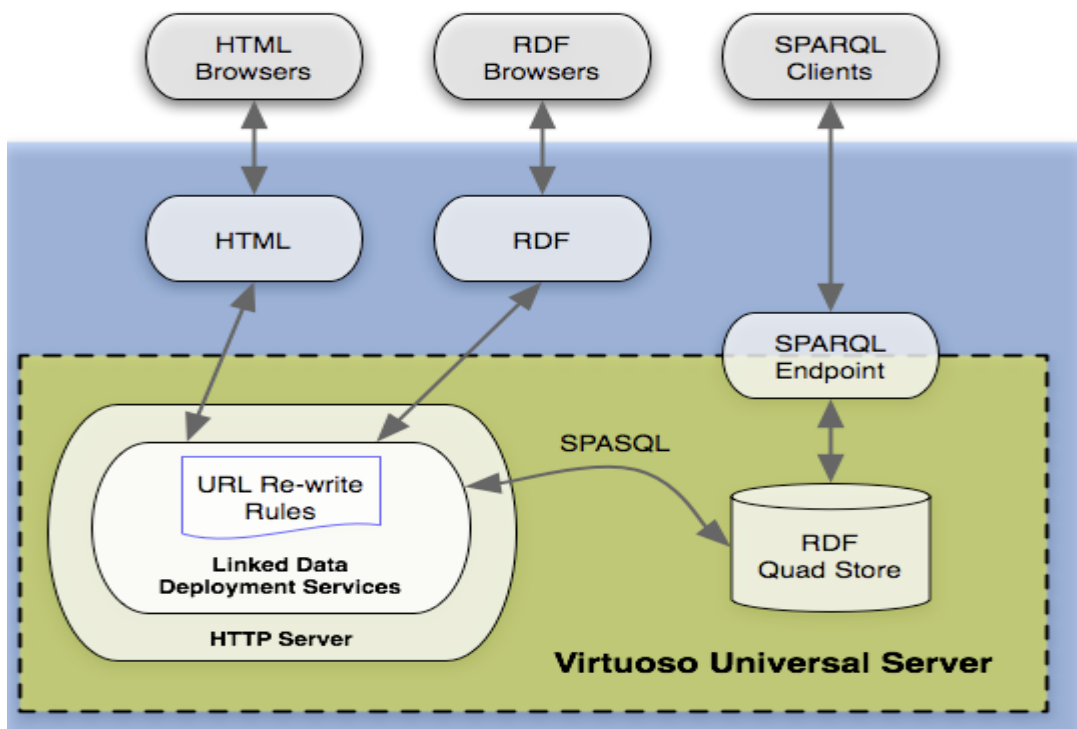
uma das maiores e mais famosas bases de RDF, além de contar com endpoint Sparql para testar as consultas.

#### 4.2.1. DBpedia

Projeto criado com o objetivo de estruturar os dados extraídos da Wikipedia, ou seja, utilizando a wikipédia como o banco de dados de informações enquanto a própria DBpedia serve como um consultor que faz a estruturação dos dados da Wikipedia e demonstra essas informações estruturadas na Web.

A DBpedia utiliza uma infraestrutura chamada OpenLink Virtuoso, como demonstrado na figura 11, onde está infraestrutura fornece o acesso aos dados RDF da DBpedia através do endpoint SPARQL, e traz as informações em diversos formatos, como HTML, XML, JSON, entre outros.

Figura 11 - Arquitetura de fornecimento de dados DBpedia atual



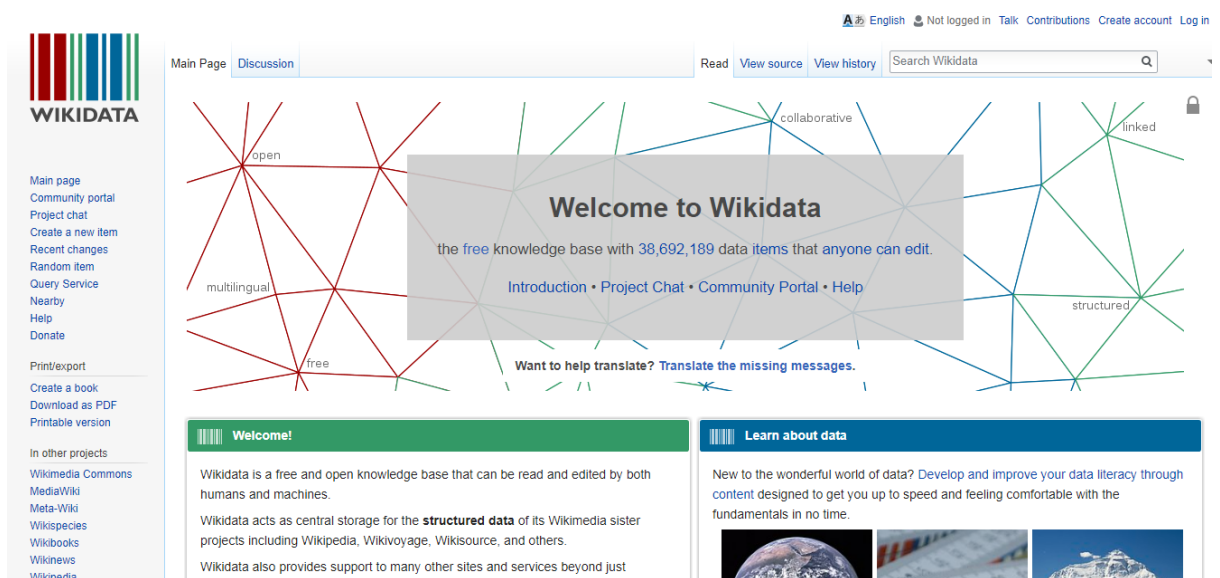
Fonte: DBpedia

Os recursos estruturados pela DBpedia são apresentados pelas URL e podem ter diversos formatos, como, por exemplo, o “<http://dbpedia.org/resource/>”. Combinando-se estas URL e o nome objeto da pesquisa, temos acesso às informações requisitadas.

## 4.2.2. Wikidata

É um projeto realizado pela Wikimedia que tem como objetivo criar uma base de conhecimento aberta, servindo como um banco de dados secundário, tendo suas principais características centradas em fornecer dados em diversos idiomas, ser colaborativa e realizar coleta de dados estruturados e dados livres, para serem copiados, modificados e compartilhados. Estes dados podem, inclusive, serem utilizados para fins lucrativos sem que o utilizador sofra algum tipo de penalidade ou punição.

Figura 12 - Site da Wikidata



Fonte: Wikidata (2017)

## 4.2.3. SIOP

Sistema Integrado de Planejamento e Orçamento (SIOP) é um sistema estruturado do governo brasileiro para informatização dos processos de Planejamento e Orçamento do Governo Federal. Através deste sistema, o governo realizou as publicações dos dados da despesa do Orçamento Geral da União no período de 2000 a 2017, obrigatório devido a Lei de Acesso à Informação (Lei 12.527, de 18 de novembro de 2011).

### 4.3. API para realização de busca RDF

Para a realização das buscas em RDF foi necessário achar, primeiramente, a API que pudesse fornecer a funcionalidade para trabalhar com SPARQL. Desta forma, localizou-se algumas API que promovem essa funcionalidade, conforme quadro abaixo. Porém, para a realização deste trabalho, foi escolhida a API Easy RDF, já que a mesma possui uma boa documentação e é altamente recomendada pelo universo de usuários que a utilizam.

Quadro 2 - Lista de API

| API         | Descrição   |
|-------------|---|
| Easy RDF    | Biblioteca para o PHP criada para facilitar as consultas e criações em RDF, desenvolvida para uso de pessoas experientes e inexperientes. Nesta biblioteca foram realizados os testes da aplicação.<br>Aplicação é capaz de criar gráficos de objeto PHP. |
| Bordercloud | API simples para a utilização do SPARQL nas aplicações.   |
| RDFIO       | API para importar e exportar RDF da Semântica MediaWiki. Também fornece o Sparql end point.   |

Fonte: Elaborado pelo próprio autor

### 4.4. Análise de Requisito Não Funcional

Elaborado quadro 3 com base nos requisitos não funcionais do protótipo, ou seja, requisitos ligados diretamente com o sistema do protótipo, conforme descrito no Capítulo 2:

Quadro 3 - Requisitos não funcionais

| Propriedade   | Sistema   |
|---------------|---|
| Usabilidade   | <ol style="list-style-type: none"> <li>1. O sistema deve possuir uma interface simples e objetiva.</li> <li>2. Rápido aprendizado.</li> <li>3. Sistema deve possuir características interativas.</li> </ol> |
| Segurança     | <ol style="list-style-type: none"> <li>1. A base de dados deve ser protegida com usuário e senha.</li> <li>2. Sistema do administrador devem ter acesso diferente dos clientes</li> </ol>                   |
| Portabilidade | <ol style="list-style-type: none"> <li>1. Deve ser comportado em todos os navegadores.</li> <li>2. Sistema deve ser responsivo para ser utilizado em outras plataformas</li> </ol>                          |



|               |   |
|---------------|---|
| Implementação | 1. Sistema deve ser desenvolvido em PHP   |
| Padrão        | 1. O sistema deve seguir modelo MVC<br>2. Sistema foi desenvolvido no framework Laravel |

Fonte: Elaborado pelo próprio autor

#### 4.5. Análise de Requisito Funcional

Elaborado quadro 4 com base nos requisitos funcionais do protótipo, ou seja, requisitos ligados à funcionalidade do protótipo, conforme descrito no Capítulo 2:

Quadro 4 - Requisitos funcionais

| Requisito Funcional | Descrição  |
|---------------------|--|
| Cadastrar Usuário   | Função do sistema para inclusão de novos usuários.                     |
| Acessar Sistema     | Possibilita o acesso ao sistema de algumas páginas.                    |
| Consultar perfil    | Visualização dos dados cadastrais do usuário.                          |
| Buscar dados        | Funcionalidade para busca dos termos na base Sparql                    |
| Consultar Mapas     | Funcionalidade para busca de informações de acordo com região do mapa. |

Fonte: Elaborado pelo próprio autor

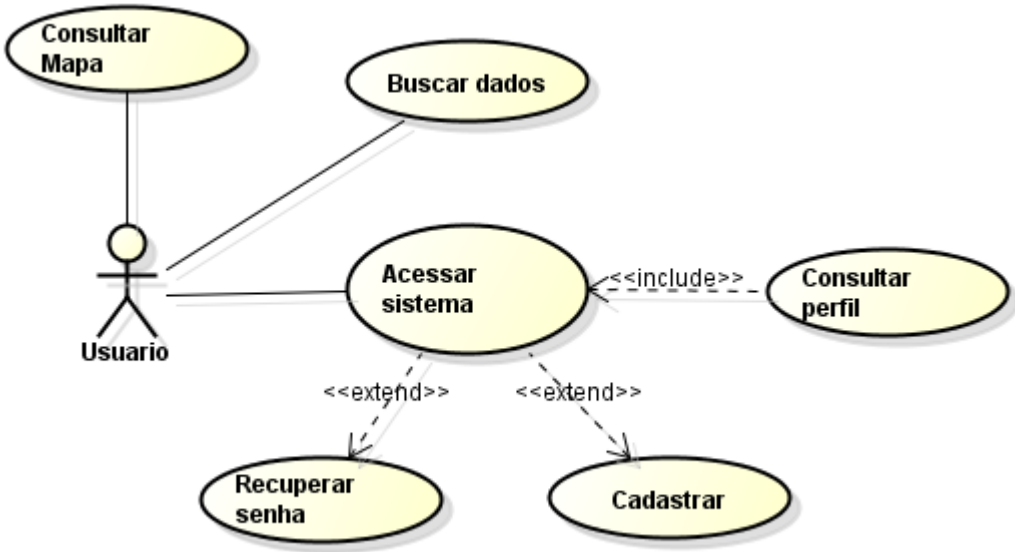
#### 4.6. Caso de Uso

O Caso de Uso é um diagrama normalmente utilizado no começo do projeto e construído juntamente com a análise de requisitos do sistema. É um diagrama simples, pois sua intenção é fazer que as pessoas mais leigas possam entender o que será feito na aplicação solicitada por eles.

Segundo Guedes (2011, p.52), o diagrama tem por objetivo “apresentar uma visão externa geral das funcionalidades que o sistema deverá oferecer aos usuários, sem se preocupar com a questão de como tais funcionalidades serão implementadas”. O diagrama pode servir como uma base para o desenvolvimento de alguns diagramas.

Com isso, para o protótipo em questão, chegou-se ao seguinte diagrama de casos de uso, conforme pode ser visto na figura 13.

Figura 13 - Caso de Uso do Protótipo



Fonte: Elaborado pelo próprio autor

4.6.1. Documentação do caso de uso

|                |  |
|----------------|--|
| Caso de Uso    | Acessar Sistema  |
| Ator Principal | Usuário  |
| Pré-condição   | Ter um registro na aplicação.                          |
| Descrição      | Este caso de uso permite ao usuário acessar o sistema. |

|                     |   |
|---------------------|---|
| Fluxo Normal        | <ol style="list-style-type: none"> <li>1. O usuário entrará no site através do navegador.</li> <li>2. Irá clicar para no botão de Login que direciona para página de Acessar sistema.</li> <li>3. Digitará as informações nos campos e pressionará logar.</li> <li>4. Será direcionado a uma página dizendo bem-vindo.</li> </ol> |
| Fluxos Exceção      | <ol style="list-style-type: none"> <li>1. Caso os dados informados não sejam os corretos, o sistema devolverá uma mensagem de erro informando “Usuário inválido”.</li> <li>2. Caso ele não informe os dados, o sistema devolverá uma mensagem de erro informando “os campos estão vazios”.</li> </ol>                             |
| Fluxos Alternativos | <ol style="list-style-type: none"> <li>1. Caso o cliente não lembre da senha, ele irá pressionar o botão de esqueceu a senha.</li> <li>2. Na página de resetar senha, ele digitará o e-mail e clicará em botão de confirmação.</li> </ol>   |
| Pós-condição        |   |

|                |   |
|----------------|---|
| Caso de Uso    | Cadastrar   |
| Ator Principal | Usuário   |
| Descrição      | Este caso de uso permite a inclusão de Usuário no banco de dados e cria as permissões de acesso ao Usuário. |

|                |   |
|----------------|---|
| Pré-condição   |   |
| Fluxo Normal   | <ol style="list-style-type: none"> <li>1. O usuário entrará no site através do navegador.</li> <li>2. Irá clicar para no botão de Registrar, que direciona para página de Cadastro.</li> <li>3. Digitará as informações nos campos e pressionará o botão de registrar.</li> <li>4. Será direcionado a uma página dizendo bem-vindo e irá logar automaticamente a pessoa na aplicação</li> </ol>   |
| Fluxos Exceção | <ol style="list-style-type: none"> <li>1. Caso o e-mail informado não seja válido, o sistema devolverá uma mensagem de erro informando “este e-mail é inválido”.</li> <li>2. Caso a senha informada não seja válida, o sistema devolverá uma mensagem de erro informando “esta senha é inválido”.</li> <li>3. Caso ele não informe os dados obrigatórios, o sistema devolverá uma mensagem de erro informando “os campos obrigatórios estão vazios”.</li> </ol> |

|                |  |
|----------------|--|
| Caso de Uso    | Consultar Perfil   |
| Ator Principal | Usuário  |
| Descrição      | Este caso de uso permite o usuário consultar as informações de seu Perfil. |
| Pré-condição   | Precisa estar logado para acessar essa página                              |

|                     |  |
|---------------------|--|
| Fluxo Normal        | <ol style="list-style-type: none"> <li>1. O usuário, após estar logado, clicará no botão de perfil e será direcionado para a página de perfil.</li> <li>2. A página perfil abrirá e mostrará todos os dados dele na tela.</li> </ol> |
| Fluxos Excepcionais | <ol style="list-style-type: none"> <li>1. Caso o usuário seja deslogado antes de clicar no perfil ,não conseguirá entrar na página de perfil.</li> </ol>   |

|                |   |
|----------------|---|
| Caso de Uso    | Consultar Mapa  |
| Ator Principal | Usuário   |
| Descrição      | Este caso de uso permite o usuário consultar os dados geográficos das regiões do mapa.  |
| Pré-condição   |   |
| Fluxo Normal   | <ol style="list-style-type: none"> <li>1. O usuário acessa o site com seu navegador.</li> <li>2. Clica no botão de Mapas e é direcionado para página do mapa.</li> <li>3. O usuário clica em alguma região do mapa.</li> <li>4. Aplicação traz as informações do mapa na página do mapa.</li> </ol> |

|                     |  |
|---------------------|--|
| Caso de Uso         | Buscar dados   |
| Ator Principal      | Usuário  |
| Descrição           | Este caso de uso permite o usuário consultar os termos   |
| Pré-condição        |  |
| Fluxo Normal        | <ol style="list-style-type: none"> <li>1. O cliente acessa o site com seu navegador.</li> <li>2. Clica no botão de Buscas e é direcionado para página da busca</li> <li>3. Aplicação abre página da busca.</li> <li>4. O usuário preenche campo de busca e envia.</li> </ol> |
| Fluxos Excepcionais | <ol style="list-style-type: none"> <li>1. Caso seja preenchido o campo com algum termo incorreto ou desconhecido, retornará uma mensagem dizendo que o “termo não foi localizado”.</li> </ol>  |

#### 4.7. Diagrama de classe

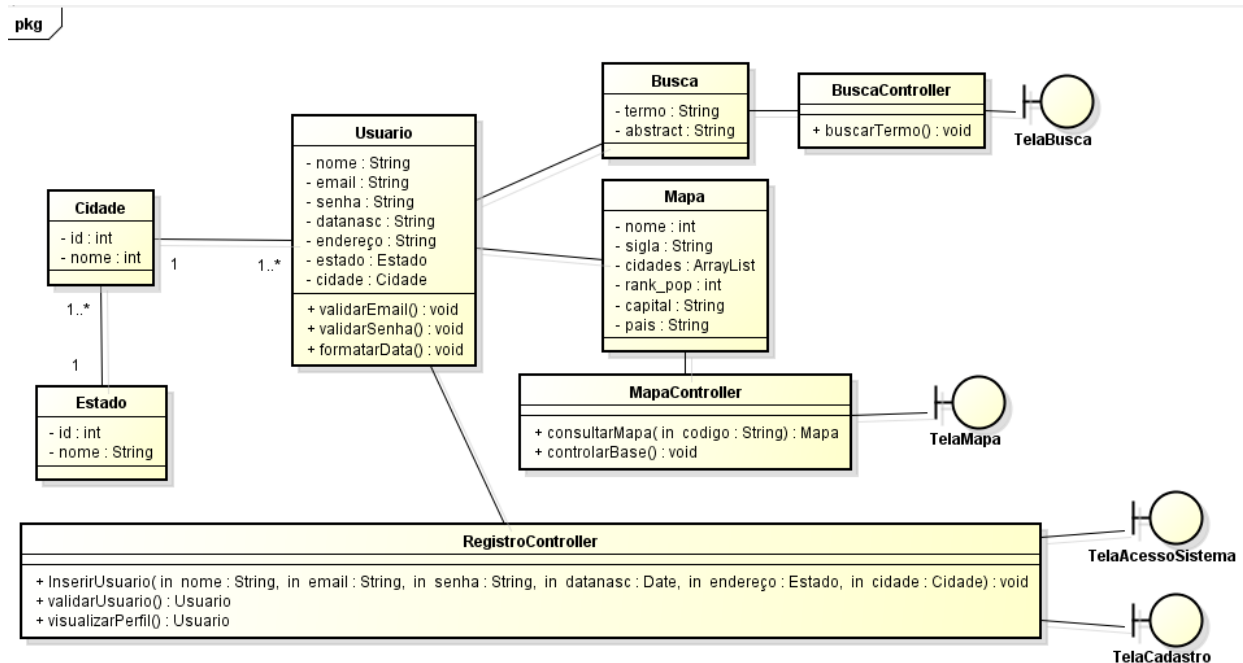
O diagrama de classe é um dos diagramas mais importantes da Unified Modeling Language(UML), pois, através dele, são mostradas as classes da aplicação, os atributos, os métodos, as relações e interações entre as classes, servindo como uma estrutura mais lógica da aplicação. É utilizado como base para construção de diversos diagramas.

O diagrama, conforme Guedes (2011, p. 101), tem como o seu principal objetivo “permitir a visualização das classes que comporão o sistema com seus respectivos atributos

e métodos, bem como em demonstrar como as classes do diagrama se relacionam, complementam e transmitem informações entre si”.

Nesta fase foi elaborado o Diagrama de Classe utilizando-se o programa Astah, para a melhor visualização da estrutura da aplicação, conforme figura 14.

Figura 14 - Diagrama de classe do Protótipo



Fonte: Elaborado pelo próprio autor

## 4.8. Desenvolvimento do Protótipo

### 4.8.1. Layout da aplicação

No desenvolvimento de aplicações web, uma das partes mais essenciais é a construção visual através do HTML e CSS. Para isto, a construção do layout do protótipo teve como base os layouts produzidos pela ferramenta Pingendo, já que o software pode gerar o código do html integrado com os atributos fornecidos pelo bootstrap, facilitando o desenvolvimento das páginas, pois o protótipo contém o bootstrap integrado, por causa do Laravel.

As figuras das páginas web que apareceram no decorrer do projeto vieram da ferramenta Pingendo e serviram como base para aplicação visual do protótipo na fase inicial.

A página inicial da aplicação não contém nenhuma funcionalidade específica, sendo voltada apenas para a apresentação da aplicação e do projeto em si. Ela contém, em seu cabeçalho, botões para redirecionar direto para sistema de login ou para sistema de cadastro e, na parte inferior, ficam os artigos e conteúdos voltados a explicar o projeto.

Figura 15 - Página Inicial

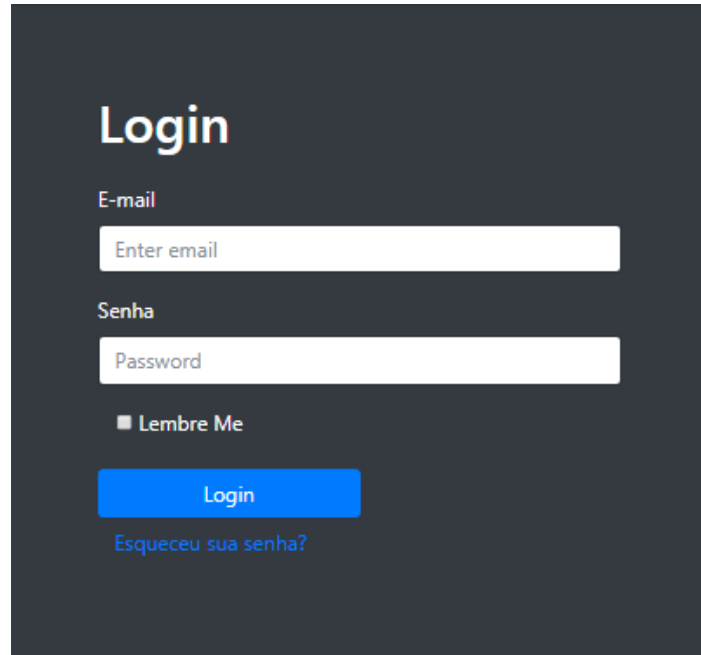


Fonte: Autoria Própria

O próximo layout faz referência ao sistema de acesso ou login da aplicação. Nesta tela, algumas funções do PHP irão interagir com o sistema de validação da senha e do e-mail. Caso tenha inserido um login incorreto, uma mensagem informando que “o usuário está incorreto” retornará para tela. No entanto, se usuário estiver correto o sistema irá avançar para a próxima tela.



Figura 16 - Acesso Sistema

A dark-themed login form with the title "Login" in white. It features two input fields: "E-mail" with a placeholder "Enter email" and "Senha" with a placeholder "Password". Below the password field is a checkbox labeled "Lembre Me". A blue "Login" button is positioned below the checkbox, and a link "Esqueceu sua senha?" is located at the bottom of the form.

**Login**

E-mail

Enter email

Senha

Password

Lembre Me

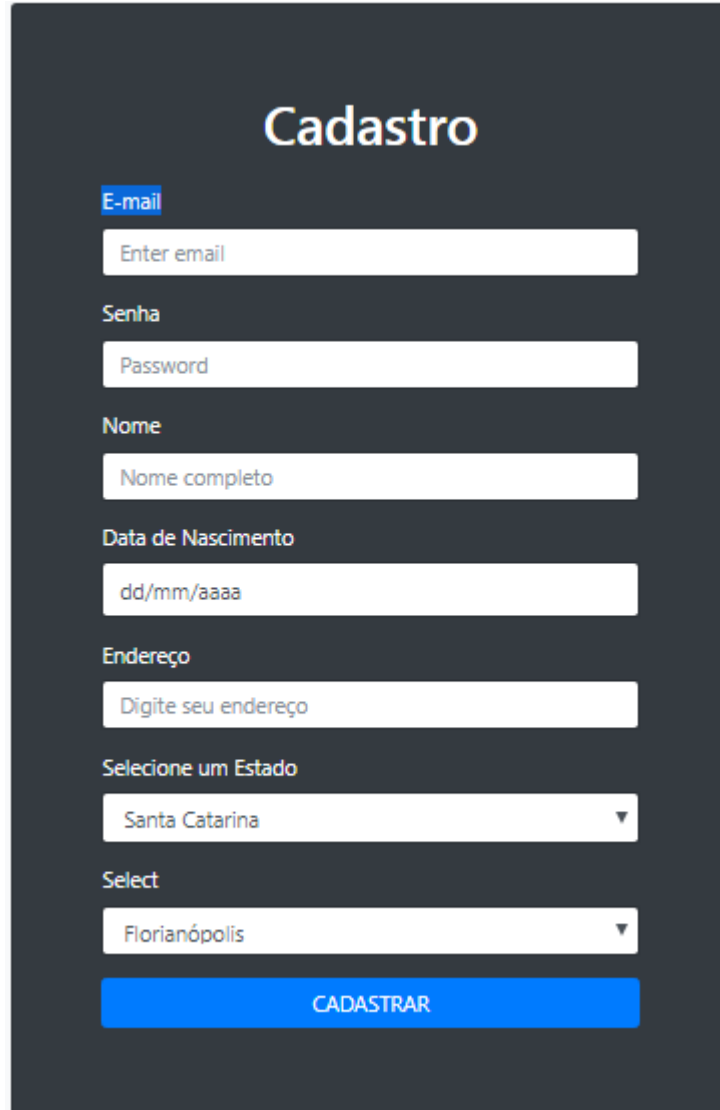
Login

[Esqueceu sua senha?](#)

Fonte: Aatoria Própria

A tela de cadastro destina-se a incluir os usuários na aplicação. Aqui, eles ganham permissão para logar no sistema. Nesta tela, também, são feitas validações de algumas informações, como e-mail e senha, que serão obrigatórios. Outras informações podem ser opcionais. No caso de alguma informação obrigatória não ser fornecida, o sistema retorna uma mensagem de erro na tela.

Figura 17 - Tela de Cadastro



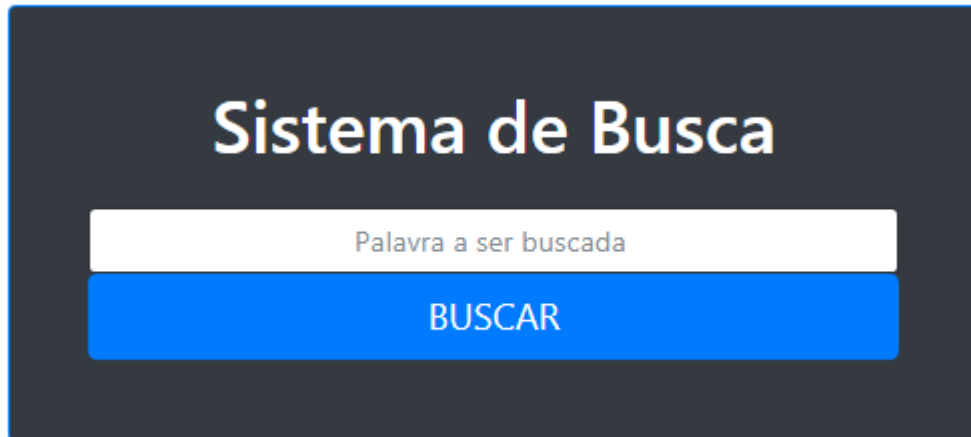
O formulário de cadastro, intitulado "Cadastro", apresenta os seguintes campos e elementos:

- E-mail:** Campo de texto com o placeholder "Enter email".
- Senha:** Campo de texto com o placeholder "Password".
- Nome:** Campo de texto com o placeholder "Nome completo".
- Data de Nascimento:** Campo de texto com o placeholder "dd/mm/aaaa".
- Endereço:** Campo de texto com o placeholder "Digite seu endereço".
- Seleção de Estado:** Menu suspenso com "Santa Catarina" selecionado.
- Seleção de Cidade:** Menu suspenso com "Florianópolis" selecionado.
- Botão de Ação:** Botão azul com o texto "CADASTRAR".

Fonte: Autoria Própria

Esta tela é destinada ao sistema de busca. Nesta fase inicial do protótipo, o layout é simples, pois a tela foi desenvolvida apenas com intenção de buscar o termo sem nenhum filtro ou definição. Nesta tela, quando o usuário digitar o termo, o sistema aciona uma busca realizada pelo SPARQL na DBpedia, que é a base escolhida para realização dos testes no protótipo, trazendo algumas informações mais comuns dos termos, como nome, descrição, tipo e alguns outros termos associados.

Figura 18 - Sistema de Busca



Sistema de Busca

Palavra a ser buscada

BUSCAR

Fonte: Autoria Própria

Esta é a tela para visualizar o mapa e os dados geográficos contidos nas unidades federativas selecionadas, como bem pode ser visto o mapa dado pela figura a seguir. O mapa é interativo e, assim, quando alguém clica em algum Estado, o sistema utiliza a consulta através do SPARQL para retornar informações vindas da base RDF, como por exemplo, a DBpedia, para preencher os campos como nome, região, capital, etc... no painel ao lado. Aqui, existe, também, um botão dropdown para selecionar a base de dados de onde são retiradas as informações. São inseridas algumas bases que já possuem informações deste conteúdo geográfico para a busca. No momento que o usuário escolher qual base será consultada, a busca trará as informações contidas e preencherá os campos. A princípio, o protótipo foi modelado desta forma para verificar a diferença de informações contidas nas bases, podendo, assim, analisar qual base está mais atualizada e quais são as informações que estão faltando em cada uma das bases disponíveis.

Figura 19 - Tela do Mapa e dados geográficos



Fonte: Autoria Própria

A construção dos layouts nesta fase do protótipo concentrou-se na simplicidade, sendo suficiente para testar algumas das funcionalidades, sem pensar muito na estética mas, sim, em sua usabilidade.

#### 4.8.2. Programação utilizando Laravel

Como mencionado anteriormente, esta aplicação foi construída utilizando o framework Laravel. Sendo assim, muitas classes e funcionalidades foram construídas automaticamente, sem a necessidade da codificação do desenvolvedor. Além do framework Laravel, algumas tecnologias foram utilizadas em conjunto para construção do protótipo, como HTML, CSS, JavaScript, PHP, MYSQL e SPARQL.

Um dos exemplos dessa automatização do Laravel é a criação das classes que tratam do cadastro e login do usuário. No momento da definição do projeto, o próprio framework já criou automaticamente a classe User para servir como a classe modelo dos usuários, além de cuidar de toda interação com banco de dados, a classe de controle e tela de cadastro e login.

Figura 20 - Classe User criada pelo Laravel no protótipo

```
<?php
namespace App;

use Illuminate\Notifications\Notifiable;
use Illuminate\Foundation\Auth\User as Authenticatable;

class User extends Authenticatable
{
    use Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'name', 'email', 'password', 'data_nascimento', 'endereço', 'cidade', 'estado',
    ];

    /**
     * The attributes that should be hidden for arrays.
     *
     * @var array
     */
    protected $hidden = [
        'password', 'remember_token',
    ];
}
```

Fonte: Autoria Própria

Para criação de classes-modelo, de controle ou do banco de dados para os modelos, o Laravel não necessita que seja criado qualquer arquivo de forma manual. Ele possui comandos específicos para construção deste tipo de classe.

O código representado na figura 21 demonstra criação do banco de dados pelo framework. E como pode ser visto, conforme o desenvolvedor vai inserindo mais da variável '\$table', mais colunas ele cria na tabela. Através das funções String e date, é que são gerados os tipos de dados para cada campo na tabela.

Figura 21 - Classe responsável pela criação das tabelas do Usuário

```
<?php
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreateUsersTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->increments('id');
            $table->string('name');
            $table->string('email')->unique();
            $table->string('password');
            $table->date('data_nascimento');
            $table->string('endereço');
            $table->string('cidade');
            $table->string('estado');
            $table->rememberToken();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('users');
    }
}
```

Fonte: Autoria Própria

A classe representada pela figura 22 é a responsável pela validação dos dados oriundos da tela de cadastro, por meio da função Validator. Nesta validação, pode-se conferir o tipo de variável, a quantidade, e se ela obrigatória ou não na execução do cadastro.

Figura 22 - Classe controle do Registro do Usuário

```

protected function validator(array $data)
{
    return Validator::make($data, [
        'name' => 'required|string|max:255',
        'email' => 'required|string|email|max:255|unique:users',
        'password' => 'required|string|min:6|confirmed',
        'data_nascimento' => 'string',
        'endereço' => 'string',
        'cidade' => 'string',
        'estado' => 'string',
    ]);
}

/**
 * Create a new user instance after a valid registration.
 *
 * @param array $data
 * @return \App\User
 */
protected function create(array $data)
{
    return User::create([
        'name' => $data['name'],
        'email' => $data['email'],
        'password' => bcrypt($data['password']),
        'data_nascimento' => $data['data_nascimento'],
        'endereço' => $data['endereço'],
        'cidade' => $data['cidade'],
        'estado' => $data['estado'],
    ]);
}
}

```

Fonte: Autoria Própria

A tela representada pela figura 23 é para mostrar o código da tela de login. Através desta figura, podemos visualizar a estrutura do HTML feito pelo Laravel, onde ele utiliza section e extends para chamar outras classes que complementam o template da página. Também mostra a criação dos formulários e ações que interagem com ele. Este mesmo tipo de formulário também pode ser encontrado na tela de cadastro, onde a única diferença percebida é relativa ao conteúdo sendo fornecido e o local para validação.

Figura 23 - Código da tela de Login

```

@extends('layouts.app')
@section('content')
<div class="py-5 bg-light">
  <div class="container">
    <div class="row">
      <div class="col-md-3"> </div>
      <div class="col-md-6">
        <div class="card text-white p-5 bg-dark">
          <div class="card-body">
            <h1 class="mb-4">Login</h1>
            <form action="{{ route('login') }}" method="POST" action="{{ route('login') }}">
              {{ csrf_field() }}
              <div class="form-group{{ $errors->has('email') ? ' has-error' : '' }}">
                <label for="email" class="col-md-4 control-label">EMAIL</label>
                <input id="email" type="email" type="email" class="form-control" placeholder="Enter email" name="email" value="{{ old('email') }}"
                  required autofocus>
                @if ($errors->has('email'))
                  <span class="help-block">
                    <strong>{{ $errors->first('email') }}</strong>
                  </span>
                @endif
              </div>
              <div class="form-group{{ $errors->has('password') ? ' has-error' : '' }}">
                <label for="password" class="col-md-4 control-label">Senha</label>
                <input id="password" type="password" class="form-control" placeholder="Password" name="password" required>
                @if ($errors->has('password'))
                  <span class="help-block">
                    <strong>{{ $errors->first('password') }}</strong>
                  </span>
                @endif
              </div>
              <div class="form-group">
                <div class="col-md-6 col-md-offset-4">
                  <div class="checkbox">
                    <label>
                      <input type="checkbox" name="remember" {{ old('remember') ? 'checked' : '' }}> Remember Me
                    </label>
                  </div>
                </div>
              </div>
              <div class="form-group">
                <div class="col-md-8 col-md-offset-4">
                  <button type="submit" class="btn text-center btn-primary btn-block">
                    Login
                  </button>
                  <a class="btn btn-link" href="{{ route('password.request') }}">
                    Forgot Your Password?
                  </a>
                </div>
              </div>
            </form>
          </div>
        </div>
      </div>
    </div>
  </div>
</section>

```

Fonte: Autoria Própria

A próximas imagens referem-se ao comando de JavaScript para criação do mapa e de suas interações. Conforme podemos observar, o JavaScript utiliza funções que são ativadas através do clique da movimentação mouse sobre o mapa. E é através dessas funções que são trazidas as informações dos estados. Para a implementação dessa funcionalidade, foi utilizada uma função já existente e diversas bibliotecas disponibilizadas pela linguagem. Como resultado do grande número de bibliotecas disponíveis, algumas inconsistências foram encontradas no mapa, chegando a prejudicar a visualização da página. Estas inconsistências foram resolvidas com a ordem correta da chamada das bibliotecas utilizadas.



Figura 24 - Código Javascript do Mapa interativo

```

<script type="text/javascript">
$(function () {

    var map_settings = {
        map: 'brazil',
        zoomButtons: false,
        zoomMax: 1,
        regionStyle: {
            initial: {
                'fill-opacity': 0.9,
                stroke: '#000',
                'stroke-width': 100,
                'stroke-opacity': 1
            },
            hover: {
                fill: '#00709a'
            }
        },
        backgroundColor: '#ffffff',
        series: {
            regions: [{
                values: {
                    // Região Norte
                    ac: '#fff9c2',
                    am: '#fff9c2',
                    ap: '#fff9c2',
                    pa: '#fff9c2',
                    ro: '#fff9c2',
                    rr: '#fff9c2',
                    to: '#fff9c2',
                    // Região Nordeste
                    al: '#fcdeeb',
                    ba: '#fcdeeb',
                    ce: '#fcdeeb',
                    ma: '#fcdeeb',
                    pb: '#fcdeeb',
                    pe: '#fcdeeb',
                    pi: '#fcdeeb',
                    rn: '#fcdeeb',
                    se: '#fcdeeb',
                    // Região Centro-Oeste
                    df: '#feb83d',
                    go: '#feb83d',
                    ms: '#feb83d',
                    mt: '#feb83d',
                    // Região Sudeste
                    es: '#e8ec9b',
                    mg: '#e8ec9b',
                    rj: '#e8ec9b',
                    sp: '#e8ec9b',
                    // Região Sul
                    pr: '#fef56c',
                    rs: '#fef56c',
                    sc: '#fef56c'
                },
                attribute: 'fill'
            }
        ]
    },
    container: $('#brazil-map'),
    onRegionClick: function (event, code) {
        var code = code;
        $('#clicked-region span').text(code);
        $.ajaxSetup({
            headers: {
                'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')
            }
        });
        jQuery.ajax({
            type: "POST",
            data: {varB: code},
            success: function(data)
            {
                window.location = "mapas?varB="+code;
            }
        });
    },
    onRegionOver: function (event, code) {
        $('#hovered-region span').text(code);
    }
});
map = new jvm.WorldMap($.extend(true, {}, map_settings));
</script>

```

Fonte: Autoria Própria

### 4.8.3. Programação da consulta SPARQL

Para programação do Sparql, foi utilizada uma biblioteca externa chamada EasyRDF. Esta biblioteca permite a utilização de diversas funções no PHP para a manipulação e realização de consultas de Sparql. Para o desenvolvimento das funcionalidades do mapa e da busca foram utilizadas as funções desta biblioteca.

Para possibilitar a busca das informações nas bases, a biblioteca do EasyRDF possui uma classe chamada *EasyRDF\_Sparql\_Client* que realiza a conexão com a base escolhida pelo desenvolvedor. Para isso, precisamos apenas informar a URL do *endpoint* da base para que possa começar a utilizar as funções de busca. Este processo é equivalente à criação da conexão com o banco de dados MySQL. Só que, ao invés de ser informado o IP de conexão e o nome database na configuração, o Sparql utiliza a URL para localizar a base na web e começar a consultar as informações.

Como exemplo, temos a figura 25, que mostra o código para a consulta do Mapa na base da DBpedia. Com o intuito de trazer as informações na tela, são utilizadas funções para a criação dos prefixos da base. Para tanto, basta colocarmos o nome que dará o prefixo e a URL de referência. Também, podemos ver que as consultas realizadas pelo Sparql são semelhantes às realizadas pelo SQL, diferenciando-se apenas em alguns pontos, como na utilização do OPTIONAL - para que a recuperação da informação possa ser opcional - ou do FILTER com LangMatches, que retorna a busca com apenas o formato de linguagem solicitada, como pt (português), visto na figura.

Figura 25 - Código para consulta das informações do Mapa

```

EasyRdf_Namespace::set('category', 'http://dbpedia.org/resource/Category:');
EasyRdf_Namespace::set('dbpedia', 'http://dbpedia.org/resource/');
EasyRdf_Namespace::set('dbo', 'http://dbpedia.org/ontology/');
EasyRdf_Namespace::set('dbp', 'http://dbpedia.org/property/');

$sparql = new EasyRdf_Sparql_Client('http://dbpedia.org/sparql');
$result = $sparql->query(
    'select distinct ?name ?Capitalname ?Sigla ?Area ?Tipo ?Rpop ?Leader ?Country where {'.
    '<http://dbpedia.org/resource/.$estado.> rdfs:label ?name .'.
    '<http://dbpedia.org/resource/.$estado.> dbp:seat ?Capital .'.
    '?Capital rdfs:label ?Capitalname .'.
    'OPTIONAL { <http://dbpedia.org/resource/.$estado.> dbp:coordinatesRegion ?Sigla .}'.
    'OPTIONAL {<http://dbpedia.org/resource/.$estado.> dbp:areaRank ?Area .}'.
    'OPTIONAL {<http://dbpedia.org/resource/.$estado.> dbo:type ?Tipo .}'.
    'OPTIONAL {<http://dbpedia.org/resource/.$estado.> dbo:populationTotalRanking ?Rpop .}'.

    '<http://dbpedia.org/resource/.$estado.> dbo:country ?Country .'.
    'FILTER (langMatches(lang(?name), "pt")) .'.
    'FILTER (langMatches(lang(?Capitalname), "en")) .'.
    '}'
);
foreach ($result as $row) {

    echo "<h4>".$row->name."</h4>\n";
    echo "<li>".$row->Capitalname."</li>\n";
    echo "<li>".$row->Sigla."</li>\n";
    echo "<li>".$row->Area."</li>\n";
    echo "<li>".$row->Tipo."</li>\n";
    echo "<li>".$row->Rpop."</li>\n";

    echo "<li>".$row->Country."</li>\n";
}
}
>>

```

Fonte: Autoria Própria

Para realização das consultas, utilizou-se o sistema Virtuoso da DBpedia,. Inicialmente, a execução foi feita no Virtuoso, até que a consulta funcionasse. Somente após isso, a consulta foi inserida no código da aplicação, já que as primeiras consultas que foram realizadas tiveram os retornos inconclusivos ou em branco, deixando muitas dúvidas sobre o que teria causado a falta de retorno ou erros SPARQL.

Para tentar solucionar-se o problema, foi realizada a consulta diretamente no endpoint da base, com o objetivo de descartar qualquer problema relacionado à biblioteca do PHP. Com isso, descobrimos que, dependendo de como é feita a busca da informação na query de consulta, basta um parâmetro que não faça sentido na busca ou que não está relacionado, na base, aos demais parâmetros, para que a mesma retorne em branco, sem dar nenhuma mensagem de erro. Por isso, deve-se prestar bem atenção no que está procurando, conferir se a busca faz sentido e se existe ligação na base de dados.

A utilização do SPARQL foi de suma importância para a finalização deste protótipo. Por essa razão, necessita-se compreender bem as buscas e informações necessárias a serem trazidas pelo protótipo. Evita-se, assim, a recuperação de informações desnecessárias ou erros que levem o Sparql a não trazer nenhuma informação.

## 4.9. Teste da aplicação

Para a realização dos testes do protótipo, utilizou-se o framework de teste PHPUnit, que realiza o tipo de teste por unidade. Este tipo teste é feito para mostrar que cada unidade do software atende à especificação proposta, ou seja, o framework apenas valida se a classe está fazendo a função que foi encarregada de fazer.

O PHPUnit serve para identificar as possíveis falhas e avisar ao desenvolvedor onde está problema. Os testes que foram realizados neste projeto foram focados na parte da classe do usuário e das páginas da aplicação, mostrando, assim, como estes testes forma feitos e os resultados gerados por ele.

A figura 26 mostrada, logo abaixo, refere-se a uma classe de teste a ser validade pelo PHPUnit. Neste caso, a classe sendo testada é a do usuário. Aqui, colocaram-se variáveis, de modo a simular um cadastro de usuário na aplicação, solicitando-se o retorno desta operação.

Figura 26 - Classe teste do PHPUnit

```
<?php
namespace Tests\Unit;

use Tests\TestCase;
use Illuminate\Foundation\Testing\RefreshDatabase;

class UserTest extends TestCase
{
    /**
     * A basic test example.
     *
     * @return void
     */
    public function testCreateUser()
    {
        $user = \App\User::create([
            'name' => "Ted",
            'email' => "teste@teste3",
            'password' => bcrypt(123456),
            'data_nascimento' => date("Y-m-d H:i:s"),
            'endereco' => "Tested", |
            'cidade' => "SC",
            'estado' => "Floirianos",
        ]);

        return $user;
    }
}
```

Fonte: Autoria Própria

A próxima figura mostra a representação de um erro na hora da validação do código PHP. Pode-se observar que PHPUnit é ativado via Prompt de Comando e que todos os resultados são demonstrados no prompt. No caso abaixo, demonstra-se um erro que está relacionado ao banco de dado, mais especificamente, acusando a divergência entre o tipo passado pela variável em PHP e o tipo do campo definido no banco de dados.

Figura 27 - Tela com os erros apontados pelo PHPUnit

```

1) Tests\Unit\UserTest::testCreateUser
Illuminate\Database\QueryException: SQLSTATE[HY000]: General error: 1364 Field 'data_nascimento' doesn't have a default value (SQL: insert into `users` (`name`, `email`, `password`, `endereco`, `cidade`, `estado`, `updated_at`, `created_at`) values (Ted, teste@teste123, $2y$10$G03zATf/Vj834ju1C4bTH.h6TkHXNirK6WUP9ghhXmp.U91UiPvdG, Tested, SC, Floirianpos, 2017-11-21 03:07:58, 2017-11-21 03:07:58))

C:\xampp\htdocs\TCC\vendor\laravel\framework\src\Illuminate\Database\Connection.php:664
C:\xampp\htdocs\TCC\vendor\laravel\framework\src\Illuminate\Database\Connection.php:624
C:\xampp\htdocs\TCC\vendor\laravel\framework\src\Illuminate\Database\Connection.php:459
C:\xampp\htdocs\TCC\vendor\laravel\framework\src\Illuminate\Database\Connection.php:411
C:\xampp\htdocs\TCC\vendor\laravel\framework\src\Illuminate\Database\Query\Processors\Processor.php:32
C:\xampp\htdocs\TCC\vendor\laravel\framework\src\Illuminate\Database\Query\Builder.php:2135
C:\xampp\htdocs\TCC\vendor\laravel\framework\src\Illuminate\Database\Eloquent\Builder.php:1270
C:\xampp\htdocs\TCC\vendor\laravel\framework\src\Illuminate\Database\Eloquent\Model.php:708
C:\xampp\htdocs\TCC\vendor\laravel\framework\src\Illuminate\Database\Eloquent\Model.php:673
C:\xampp\htdocs\TCC\vendor\laravel\framework\src\Illuminate\Database\Eloquent\Model.php:536
C:\xampp\htdocs\TCC\vendor\laravel\framework\src\Illuminate\Database\Eloquent\Builder.php:753
C:\xampp\htdocs\TCC\vendor\laravel\framework\src\Illuminate\Support\helpers.php:1035
C:\xampp\htdocs\TCC\vendor\laravel\framework\src\Illuminate\Database\Eloquent\Builder.php:754
C:\xampp\htdocs\TCC\vendor\laravel\framework\src\Illuminate\Database\Eloquent\Model.php:1440
C:\xampp\htdocs\TCC\vendor\laravel\framework\src\Illuminate\Database\Eloquent\Model.php:1457
C:\xampp\htdocs\TCC\tests\Unit\UserTest.php:25

Caused by
PDOException: SQLSTATE[HY000]: General error: 1364 Field 'data_nascimento' doesn't have a default value

```

Fonte:

Fonte: Autoria Própria

Após o reparo do erro na classe de teste do Usuário, fazendo com que a variável da data de nascimento fosse fornecida corretamente, a transmissão da mesma ao banco de dados ocorreu com sucesso. Nesta situação, o PHPUnit mostrou a tela representada pela figura abaixo.

Figura 28 -Tela informando que o teste deu certo.

```
C:\xampp\htdocs\TCC>vendor\bin\phpunit
PHPUnit 6.4.3 by Sebastian Bergmann and contributors.

..R                                                    3 / 3 (100%)

Time: 394 ms, Memory: 10.00MB

There was 1 risky test:

1) Tests\Unit\UserTest::testCreateUser
This test did not perform any assertions

OK, but incomplete, skipped, or risky tests!
Tests: 3, Assertions: 2, Risky: 1.
```

Fonte: Autoria Própria

Com o PHPUnit é possível realizar diversos tipos de teste em sua aplicação, aumentando, assim, a confiabilidade nos códigos e a segurança, tendo em vista a diminuição de falhas. Os testes realizados, porém, dependem bastante do enfoque que o desenvolvedor quer dar a diferentes módulos da sua aplicação.

#### 4.10. Implementações Futuras

Como este projeto tratou de um protótipo de aplicação, o conjunto completo de funcionalidades não está concluído e pode ser estendido para, futuramente, finalizar sua construção.

Vale frisar, também, que mudanças, adaptações e correções em seu código necessitam ser implementadas, a fim de que o mesmo possa ser disponibilizado para utilização completa.

Neste capítulo, abordamos algumas funcionalidades sugeridas e adicional aprimoramento do que se produziu até o momento, com o propósito de explicar estas funções adicionais e fornece um melhor entendimento de como a aplicação pode ser utilizada.

##### 4.10.1. Implementações de uma Sistema Geográfico (SIG)

Como pode ser observado no protótipo, uma das funções que foi implementada no software é voltada à busca de informações para regiões do mapa através das bases RDF. Porém, para aprimorar esta funcionalidade, sugere-se, em uma futura revisão, que a

busca seja estendida a um mapa-múndi, abrangendo todas as informações provindas das bases RDF referentes à região selecionada pelo usuário.

Esta funcionalidade se destinaria não somente a visualizar os dados mais básicos da região, mas também, retornaria informações mais complexas. Estas, por sua vez, são geralmente utilizadas por organizações, principalmente do setor do Geoprocessamento. As informações retornadas pelo sistema seriam utilizadas para gerar gráficos e estatísticas.

#### **4.10.2. Busca aprimorada**

O protótipo foi construído com a intenção de facilitar a busca e visualização das informações provindas das bases de RDF, mas como o protótipo se encontra em sua fase inicial, sua capacidade de busca está ainda muito limitada a algumas bases, trazendo apenas algumas informações resultantes da busca executada pelo SPARQL.

A busca poderia ser atualizada para que possa ser feita em diversas bases RDF disponíveis, ampliando assim, a área de procura do termo, e trazendo consigo, todos os tipos de dados possíveis e que estão ligados ao termo buscado. Adicionalmente, poderia se pensar em um sistema para filtrar a busca por segmentos, como pessoas, objetos, lugares e outros - para o caso de o usuário desejar uma busca específica de seu termo.

#### **4.10.3. Sistema para baixar os arquivos**

O protótipo até o momento não possui esta característica, permitindo apenas trazer as informações a serem visualizadas pelo o usuário. Futuramente, a aplicação poderia contar com uma funcionalidade que permitisse baixar as informações buscadas pelos usuários para, assim, possibilitar uma maior manipulação das informações e disponibilizá-las em diversos formatos. Isto daria maior praticidade ao usuário na utilização de seus afazeres.

Apesar de se poder estender o leque de formatos que a aplicação poderia baixar e manipular, o foco seria voltado apenas para distribuição de formatos abertos ou com licenças abertas como CSV, PDF, XML, HTML, RDF e muitos outros.

## 5. CONCLUSÕES

No projeto desenvolvido foi construído um protótipo de aplicação na web voltado à busca de dados abertos conectados, que pudesse extrair os dados das bases, fazendo com que, assim, pudéssemos utilizar as informações de forma eficiente, prática e rápida. Isso facilitaria, em muito, diversas atividades relacionadas ao tratamento e ao uso da informação, com reflexo direto em empresas e instituições dos mais variados tipos. Por exemplo, um gestor de uma empresa poderia utilizar a base de dados orçamentários, legislação trabalhista em vigor, normas fiscais vigentes, e outros, e utilizar este conhecimento diretamente na administração do seu plano de negócios.

Ademais, como já foi explicitado anteriormente, com a possível melhora das funções do protótipo, podemos disparar consultas de forma a atingir uma ampla base de dados RDF, trazendo informações dos mais variados tipos, integradas e correlacionadas, que poderiam ser utilizadas, dependendo do caso, para a tomada de decisão e gestão da informação, a critério de quem as fosse utilizar.

O tópico relacionado à pesquisa bibliográfica permitiu que houvesse um aumento do entendimento da parte conceitual da aplicação, além de mostrar a importância dos dados conectados para a sociedade. Mostrou, também, tecnologias que poderiam ser utilizadas para construção dessa aplicação de forma mais eficiente.

A construção desse protótipo possibilitou trabalhar e exercitar, aplicando, na prática, diversos conhecimentos ministrados durante o curso CSTGTI, como análise de sistemas, projetos, modelagem do conhecimento, gestão do conhecimento, programação web, banco de dados, Interação Humano Computador, entre outros.

O conceito de web semântica foi fundamental para o entendimento da problemática causada pelo volume gigantesco de dados disponíveis, hoje, na internet, e o caos que representa a busca e acesso destes mesmos dados. Desta forma, a web semântica procura, por meio da atribuição de significado bem definido a cada informação, organizar e facilitar sua recuperação, permitindo uma melhor interação com o usuário.

O protótipo, apesar de ainda estar na parte inicial, poderá, com o decorrer do tempo, beneficiar as pessoas – tanto leigos quanto profissionais – com a busca de dados nas bases RDF, permitindo, assim, que, de forma prática, rápida e acessível, tenha-se a



informação nas pontas dos dedos, sem a necessidade de se garimpar, manualmente, os milhões de terabytes de dados que a web hoje nos oferece.

## REFERÊNCIAS

- Almeida, Maurício Barcellos. **Uma introdução ao XML, sua utilização na Internet e alguns conceitos complementares**. Brasília, 2002. v. 31, n. 2, p. 5-13.
- Bezerra, Eduardo. **Princípios de análise e projetos de sistemas com UML**. Editora Elsevier. Rio de Janeiro. 2017.
- BIZER, C.; Heath, T. and Berners-Lee, T. **Linked data** – the story so far, International Journal on Semantic Web & Information Systems, p. 9, 2009.
- BREITMAN, Karin. **Web Semântica: a Internet do futuro**. Rio de Janeiro: LTC, p. 7, 2005.
- CAMPOS, Linair Maria et al. **Dados abertos interligados e o espaço do profissional de informação: Uma aplicação no domínio da enfermagem**. 2013.
- CONGLIAN, Caio Saraiva; SEGUNDO, José Eduardo Santarem. **Europeana no Linked Open Data: conceitos de Web Semântica na dimensão aplicada das Humanidades Digitais**. Encontros Bibli: revista eletrônica de biblioteconomia e ciência da informação, v. 22, n. 48, p. 90, 2016.
- DE LIMA, Júnio César; DE CARVALHO, Cedric L. **Resource description framework (rdf)**. Technical report, Universidade Federal de Goiás, 2005.
- Dziekaniak, G.; Kirinus, J. **Web semântica**. Encontros Bibli: revista eletrônica de biblioteconomia e ciência da informação, v. 9, n. 18, p. 21, 2007.
- ISO. **ISO 9241-11: ergonomic requirements for office work with visual display terminals (VDTs)** – guidance on usability. Genebra, 1998.
- Eris, Diego; Ferreira, Elcio. **HTML5 e CSS3: com farinha e pimenta**. Publicadora Tableless, São Paulo 2012. p. 27-32.
- Flanagan, David. **JavaScript: O guia definitivo**. 6° edição. Bookman Companhia Ed, 2013. p 1-2.
- GIL, Antonio Carlos. **Metodos e tecnicas de pesquisa social**. 6° edição. São Paulo: Editoras Atlas S A, p. 27. 2008.
- GIL, Antonio Carlos. **Como Elaborar Projetos de Pesquisa**. 4° edição. São Paulo: Editoras Atlas S A, p. 44-54. 2008.
- Guedes, Gilleanes T. A.; UML 2: Uma abordagem prática. 2° Edição. São Paulo: Novatec. 2011
- Isotani, Seiji; Bittencourt, Ig Ibert. **Dados Abertos Conectados**. São Paulo: Novatec, 2015.

Laufer, Carlos. **Guia de web semântica**. Governo do Estado de São paulo.p 33. 2015.  
**Linguagem de programação**. Disponível em: <[https://pt.wikipedia.org/wiki/Linguagem\\_de\\_programa%C3%A7%C3%A3o](https://pt.wikipedia.org/wiki/Linguagem_de_programa%C3%A7%C3%A3o)>. Acesso em: 28 maio. 2017.

Lopes, Yayiane dos Santos de Freitas. **Ontologia como interface de apresentação de resultados de busca**: uma proposta baseada no modelo espaço vetorial. .2017  
 <[https://repositorio.unesp.br/bitstream/handle/11449/151715/lopes\\_tsf\\_me\\_mar.pdf?sequence=3](https://repositorio.unesp.br/bitstream/handle/11449/151715/lopes_tsf_me_mar.pdf?sequence=3)>

Niederauer, Juliano: **Desenvolvendo Websites com PHP**. 2º edição. São Paulo. Editora Novatec, 2011, pág 24-26.

NIELSEN, Jakob. Usability 101: Introduction to Usability. Disponível em: <<https://www.nngroup.com/articles/usability-101-introduction-to-usability/>>. Acesso em: 28 maio 2017.

**MANUAL dos dados abertos: desenvolvedores**. São Paulo: Comitê Gestor da Internet no Brasil, 2011. Cooperação técnica científica entre Laboratório Brasileiro de Cultura Digital e o Núcleo de Informação e Coordenação do Ponto BR (NIC.br), 2011. Disponível em: <[http://www.w3c.br/pub/Materiais/PublicacoesW3C/Manual\\_Dados\\_Abertos\\_WEB.pdf](http://www.w3c.br/pub/Materiais/PublicacoesW3C/Manual_Dados_Abertos_WEB.pdf)>. Acesso em: 28 maio 2017.

**Manual do PHP**. The PHP Group. Disponível: [https://secure.php.net/manual/pt\\_BR/intro-what-is.php](https://secure.php.net/manual/pt_BR/intro-what-is.php). Acesso em: 06 novembro 2017.

MORAIS, Edison Andrade Martins; AMBRÓSIO, Ana Paula L. **Ontologias**: conceitos, usos, tipos, metodologias, ferramentas e linguagens. Universidade Federal de Goiás, 2007.

Ramalho, Rogério Aparecido Sá; Vidotti. Silvana Aparecida Borsetti Gregorio; Fujita, Mariângela Spotti Lopes. **Web Semântica**: Aspectos interdisciplinares para a organização e recuperação de informações. Florianópolis. ENCONTRO NACIONAL DE PESQUISA EM CIÊNCIA DA INFORMAÇÃO (ENANCIB), p 5, 2005.

Santos, Fernando Chagas. De Carvalho, Cedric Luiz. **Aplicações de Suporte à Web Semântica**. Relatório Técnico: Universidade Federal de Goiás. 2007

Schiessl, Marcelo. **Lexicalização de Ontologias**: o relacionamento entre conteúdo e significado no contexto da Recuperação da Informação. 2015. 261 p. Tese (Doutorado – Doutorado em Ciência da Informação) — Universidade de Brasília.

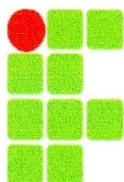
SILVA, Edna Lúcia da; MENEZES, Estera Muszkat. **Metodologia da Pesquisa e Elaboração de Dissertação**. 4º Edição. Florianópolis. UFSC. p. 20 .2005

Sommerville, Ian. **Engenharia de Software**. 9º edição, São Paulo, Pearson Prentice Hall. Disponível em:

<[http://www.ifc-camboriu.edu.br/~catia/IA16/Engenharia\\_Software\\_3Edicao.pdf](http://www.ifc-camboriu.edu.br/~catia/IA16/Engenharia_Software_3Edicao.pdf)>. Acesso em: 29 outubro 2017.

WILLEMANN, David Pedro; IBARRA, Gustavo Bestetti. **Trabalho de Conclusão de Curso Framework Java de Apoio ao Desenvolvimento de Aplicações Web com Banco de Dados, utilizando Struts, Tiles e Hibernate**. Universidade Federal de Santa Catarina, 2007.

## APÊNDICE F – Declaração de finalização de trabalho de curso



INSTITUTO FEDERAL  
SANTA CATARINA

MINISTÉRIO DA EDUCAÇÃO  
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA  
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA  
CAMPUS FLORIANÓPOLIS

### DECLARAÇÃO DE FINALIZAÇÃO DE TRABALHO DE CURSO

Declaro que o(a) estudante Alexandre Andrade Minichiello,  
matrícula nº 151005247-0, do Curso Superior de  
Gestão de Tecnologia da Informação, defendeu o trabalho intitulado  
Protótipo de aplicação web para dados abertos conectados  
o qual está apto a fazer parte do banco de dados da Biblioteca Hercílio Luz do Instituto Federal de  
Santa Catarina, Câmpus Florianópolis.

Florianópolis, 21 de dezembro de 2017.

Yerval Daminielli  
Prof. Orientador do TCC