

Universidade do Minho

**MESTRADO INTEGRADO DE ENGENHARIA
INFORMÁTICA**

**PROCESSAMENTO E REPRESENTAÇÃO DE CONHECIMENTO (2º SEMESTRE
- 2020/21)**

Relatório PRC - TP

A85813 António Alexandre Carvalho Lindo

A85400 Nuno Azevedo Alves da Cunha

A85919 Pedro Dias Parente

Braga

28 de junho de 2021

Conteúdo

1	Introdução	3
2	Ontologia	4
2.1	Classes	4
2.2	Object Properties	4
2.3	Data Properties	5
3	Arquitetura de Sistema	6
4	API	7
4.1	Comunicação com a base de dados	8
5	Front end	9
5.1	Estrutura do website	9
5.2	Vue router	9
6	Conclusão	10

Capítulo 1

Introdução

Este trabalho teve como objetivo a construção de uma ontologia sobre animais e as suas várias características, a implementação de uma API em NodeJS utilizando a base de dados ontológica GraphDB e um frontend em Vue que a consome mostrando os vários animais organizados pela sua informação.

Capítulo 2

Ontologia

De modo a construir a ontologia, começamos por estudar a informação disponível no website A-Z animals (<https://a-z-animals.com/>), uma vez que os dados utilizados para povoar a ontologia foram retirados deste website utilizando webscraping em Python. Feito isto criámos uma ontologia que abrange todos os conceitos e informações principais do website.

2.1 Classes

As classes da ontologia são as seguintes:

- Animal: Classe que representa os animais. Tem as seguintes subclasses:
 - Mammal: Representa os mamíferos;
 - Bird: Representa os pássaros;
 - Fish: Representa os peixes;
 - Insect: Representa os insetos;
 - Invertebrate: Representa os invertebrados;
 - Amphibian: Representa os anfíbios;
 - Reptile: Representa os répteis;
- Classification: Classe que representa as classificações dos animais. Tem as seguintes subclasses:
 - Classe: Representa a classe do animal;
 - Family: Representa a família do animal;
 - Genus: Representa o género do animal;
 - Kingdom: Representa o reino do animal;
 - Order: Representa a ordem do animal;
 - Phylum: Representa o filo do animal;
- Location: Representa a(s) localização(ões) do animal;

2.2 Object Properties

As Object Properties da ontologia são as seguintes:

- Animal livesIn Location (inversa: hasAnimal): Relaciona o animal com a sua localização e vice-versa;
- Animal hasPrey Animal (inversa: hasPredator): Relaciona o animal com as suas presas (e predadores);

- **Animal hasClassification Classification:** Relaciona o animal com a sua classificação (inversa: **hasAnimal**). Tem as seguintes sub propriedades:
 - **Animal hasClasse Classe:** Relaciona o animal com a sua classe;
 - **Animal hasFamily Family:** Relaciona o animal com a sua família;
 - **Animal hasKingdom Kingdom:** Relaciona o animal com o seu reino;
 - **Animal hasGenus Genus:** Relaciona o animal com o seu gênero;
 - **Animal hasOrder Order:** Relaciona o animal com a seu ordem;
 - **Animal hasPhylum Phylum:** Relaciona o animal com o seu filo;

2.3 Data Properties

As Data Properties são as seguintes:

- **colour:** Cor(es) do animal;
- **diet:** Tipo de alimentação do animal;
- **fact:** Facto sobre o animal;
- **gestationPeriod:** Período de gestação do animal;
- **habitat:** Habitat do animal;
- **image:** Imagem do animal;
- **length:** Tamanho do animal;
- **lifeSpan:** Tempo médio de vida do animal;
- **scientificname:** Nome científico do animal;
- **skinType:** Tipo de pele do animal;
- **wingspan:** Envergadura das asas do animal;
- **locationImage:** Imagem da localização do animal no mapa mundo.

Capítulo 3

Arquitetura de Sistema

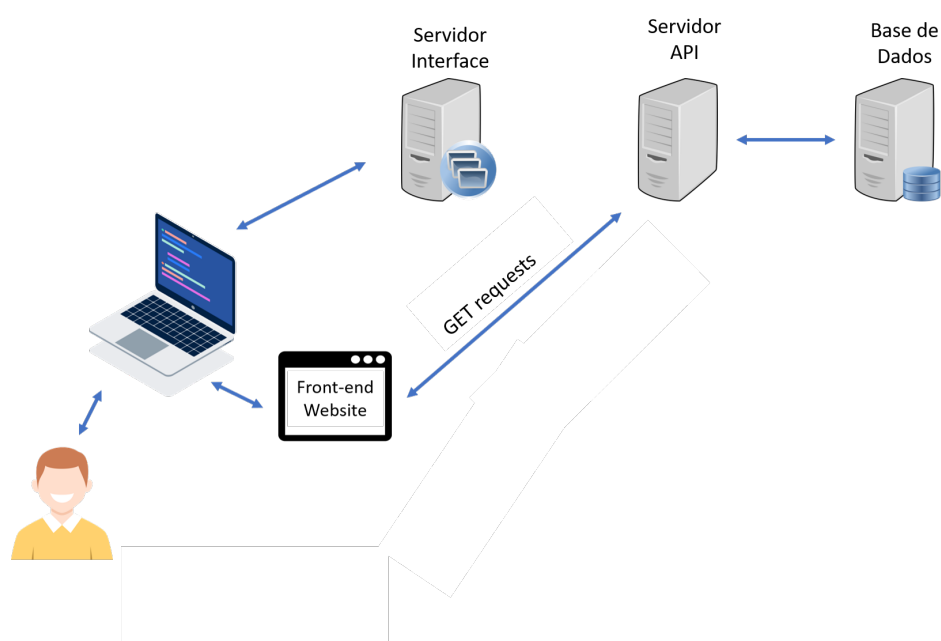


Figura 3.1: Arquitetura Aplicacional

Capítulo 4

API

A API de dados foi desenvolvida em NodeJS. Esta conecta-se com o graphDB, onde está armazenada a informação e envia queries para este executar e devolver os resultados. As rotas da API são as seguintes:

- /animals
 - GET /animals/locations: Devolve todas as localizações existentes;
 - GET /animals/types: Devolve todos os tipos de animais existentes;
 - GET /animals: Devolve todos os animais existentes. Pode ter as seguintes query strings:
 - * ?type: Tipo do animal;
 - * ?location: Localização do animal;
 - * ?Kingdom: Reino de Animal;
 - * ?Family: Família do animal;
 - * ?Classe: Classe do animal;
 - * ?Genus: Género do animal;
 - * ?Order: Ordem do animal;
 - * ?Phylum: Filo do animal;
 - * ?classification: Classificação do animal;
 - GET /animals/:animal: Devolve a informação de um dado animal passado como parâmetro;
 - PUT: /animals/:animal: Edita a informação de um dado animal passado como parâmetro;
 - DELETE: /animals/:animal: Elimina toda a informação de um dado animal passado como parâmetro;
- /classifications
 - GET /classifications/:id: Devolve todas as instâncias da classificação passada como argumento.
- /users
 - POST /users/login: Efetua o login de um administrador;
 - GET /users/logout: Efetua o logout de um administrador;
 - POST /users/registar: Regista um administrador na plataforma

Consideramos que quem usa o website são apenas utilizadores comuns que apenas podem visualizar a informação, e que para fazer alterações à ontologia é preciso fazer login pela API (ou seja, é preciso ter uma autenticação válida, sendo, por exemplo, um pesquisador) e consequentemente usar a API para fazer as alterações.

4.1 Comunicação com a base de dados

O grupo escolheu abstrair a comunicação com a base de dados presente nos controladores através do modulo *utils/graphdb.js*.

Este modulo expõe essencialmente duas funções: *execQuery()* e *execTransaction()*. A primeira para correr queries de fetching e a segunda para efectuar alterações na base de dados. Elas são uteis para o grupo não ter que se preocupar com URL encodings e prefixos ontológicos.

Está ainda presente uma função para dar parse aos resultados recebidos do GraphDB

Capítulo 5

Front end

O front end da aplicação foi desenvolvido com a framework *Vue* e os pluggins *Vuetify* e *Vue Router*.

5.1 Estrutura do website

O website está organizado em 4 views principais

- **Homepage** - A página de acesso ao website na url `"/` contém links de navegação para utilização do website.
- **Listagem de animais** - Esta página apresenta a lista dos animais disponiveis de acordo com a filtragem apresentada na query string na url `"/animals"`
- **Página de Animal** - Aqui é apresentada toda a informação relativa a cada animal, aparecem links também para listas de animais relacionados bem com link diretos para predadores e presas.
- **Pesquisa avançada** - Esta página é destinada a fazer pesquisas mais complexas do que aquelas geradas pelos menus do website

Existe ainda uma pagina para listagem das classificações, que lista por exemplo todas as familias de animais presentes na base de dados.

A aplicação foi desenvolvida como uma single page application no sentido que a página nunca é recarregada apenas são atualizados os elementos do DOM mesmo quando ocorre uma mudança de rota

5.2 Vue router

Este modulo foi utilizado para simplificar a mudança dos componentes a renderizar de acordo com a rota através que um objecto javascript que associa a cada url um componente Vue, é depois usado um componente *router-view* que trata de renderizar de acordo com esse objecto.

Este modulo tem ainda a grande vantagem de disponibilizar o objecto *\$route* que permite na paret do sript do vue ter acesso à querystring e aos parâmetros da URL

Capítulo 6

Conclusão

Em conclusão, pensamos que construímos um website com funcionalidades essenciais para o funcionamento do mesmo. O website também tem uma User Interface bastante intuitiva e natural que permite que o utilizador se "perca" a usar o site e passe muito tempo a aprender sobre os animais.

O grupo considera também que ontologia foi construída de uma forma que se adequou à sua fonte de dados mas que permite também a sua expansão não só com novos animais mas também com novos tipos e relações entre eles.

Em futuras iterações do trabalho, gostaríamos de implementar a funcionalidade de login no frontend do website, pois neste momento é pouco prático ter a funcionalidade de alteração da ontologia apenas por uso da API.