

UNIVERSIDADE DO MINHO
**Mestrado Integrado em Engenharia
Informática**
*Processamento e Representação de
Conhecimento*

World of Warcraft - Journal

Relatório

Alexandre Teixeira — A73547

Braga, Junho de 2019

Resumo

Inicialmente, é feita uma **definição do âmbito** do projeto, que explica os objetivos do mesmo.

De seguida, é feita a especificação dos processos pretendidos, de forma a esclarecer e exemplificar claramente o fluxo do software a implementar. Além disso, com o mesmo objetivo, é apresentada a modelação da ontologia, desta maneira, é possível obter uma noção precisa sobre a ligação entre os dados.

De imediato, são definidos os componentes necessários para o funcionamento do projeto:

- Ontologia
- BackEnd - Servidor Rest em *NodeJS*
- FrontEnd - Cliente em *VueJS*

Finalmente, é boa prática, referir as considerações aferidas com este projeto, de forma a melhor o caracterizar.



Conteúdo

1 Definição do Âmbito	3
1.1 Objetivos e motivação	3
1.2 Definição da Ontologia	3
1.2.1 Dados Obtidos	3
1.2.2 Classes	4
1.2.3 Propriedade de relacionamento, Object Properties	5
1.2.4 Modelo Físico	5
2 Implementação do Back-End	6
2.1 Estrutura do servidor	6
2.2 Rotas de acesso	6
3 Interface da Aplicação	8
3.1 Página Home	8
3.2 Página de Fração	8
3.3 Página de Raça	9
3.4 Modal de Spell	9
3.5 Página de Class	10
3.6 Lista de Spells	10
3.7 Lista de Zonas	11
3.8 Pagina de Zonas	11
3.9 Modal de Boss	12



Lista de Tabelas

1	Tabela de Classes	4
2	Tabela de Object Properties	5
3	Tabela de Rotas - parte 1	6
4	Tabela de Rotas	7

Lista de Figuras

1	Exemplo do Grafo gerado por GraphDB	5
2	Árvore da estrutura dos ficheiros do servidor	6
3	Página Home	8
4	Página de Fração	8
5	Página de Raça	9
6	Modal de Spell	9
7	Página de Class	10
8	Lista de Spells	10
9	Lista de Zonas	11
10	Página de Zona	11
11	Modal de Boss	12



1 Definição do Âmbito

1.1 Objetivos e motivação

O **Warcraft** é a história de um Universo fictício, no qual se desenrolam a série de livros e jogos para computador com o mesmo nome e publicados pela **Blizzard Entertainment, Inc.**

A primeira aparição deste Universo aconteceu no jogo **Warcraft: Orcs Humans** de 1995, cuja história centrava-se numa batalha de poder, entre humanos e orcs em Azeroth.

Desde esta altura, surgiram vários títulos como, **Tides of Darkness, Reign of Chaos, Defense of the Ancients**, mais conhecido como DOTA e por fim **World of Warcraft**. Após o lançamento deste título, em 2005, foram publicadas várias expansões, e com isto, surgiram algumas plataformas de suporte. Sendo neste último que este trabalho se insere, onde o principal objetivo é demonstrar a história deste Mundo.

1.2 Definição da Ontologia

1.2.1 Dados Obtidos

Através do url <http://develop.battle.net>, que fornece uma API de dados sobre os seus múltiplos jogos, foi possível obter um conjunto elevado de dados, em JSON, sobre World of Warcraft.

Com estes dados, foi necessária a criação de conversores em JavaScript, para transformar JSON em Turtle, e com isto filtrar os campos relevantes. Atentemos ao exemplo:

```
"zones": [
    {
        "id": 209,
        "name": "Shadowfang Keep",
        "urlSlug": "shadowfang-keep",
        "description": "Looming over Pyrewood.",
        "location": {
            "id": 130, "name": "Silverpine Forest"
        },
        "expansionId": 0,
        "numPlayers": "5",
        "isDungeon": true,
        "isRaid": false,
        "advisedMinLevel": 17,
        "advisedMaxLevel": 60,
        "advisedHeroicMinLevel": 85,
        "advisedHeroicMaxLevel": 85,
        "availableModes": [
            "DUNGEON_NORMAL",
            "DUNGEON HEROIC"
        ],
        "floors": 7,
        ...
    }
]
```



1.2.2 Classes

Nesta secção vão ser especificadas as diferentes classes necessárias para armazenar a informação que o sistema engloba. Para tal, após uma análise dos dados obtidos, foram identificadas as entidades abaixo apresentadas, como necessárias para suportar todos os dados gerados pelo sistema.

Class	SubClass de	Descrição
Faction	owl	Representação das diferentes frações de heróis
Alliance	Faction	Representação uma da fração
Horde	Faction	Representação uma da fração
Location	owl	Localização de zonas
Mode	owl	Modos de jogo disponíveis
NPC	owl	Representação de "non playabel characters"
Boss	NPC	Personagens inimigas presentes em zonas
PowerType	owl	Tipo de poder dos heróis
Race	owl	Representação das possíveis raças de heróis
Role	owl	Tipo de posição de cada herói
Spec	owl	Conjunto de atributos de herói
Spell	owl	Habilidade de herói
Mount	Spell	Habilidades de transporte
Talent	Spell	Habilidade de um conjunto de spec
Trait	Spell	Habilidade de Raça
WoW_Class	owl	Definição de herói
Zone	owl	Zonas de combate
Dungeon	Zone	Pequena zona de combate
Raid	Zone	Grande zona de combate

Tabela 1: Tabela de Classes



1.2.3 Propriedade de relacionamento, Object Properties

Após identificadas todas as classes, é necessário identificar os seus relacionamentos.

Na tabela seguinte são apresentados todos os relacionamentos, assim como o domínio e o contradomínio.

Propriedade	Domínio	Contradomínio	Inversa
hasCapital	Faction	Location	isCapitalOf
hasFaction	Race	Faction	isFactionOf
hasMode	Zone	Mode	isModeOf
hasNPC	Zone	NPC	isNPCOf
hasOrigin	Race	Location	isOriginOf
hasPowerType	WoW_Class	PowerType	isPowerTypeOf
hasRole	Spec	Role	isRoleOf
hasSharedTalent	WoW_Class	Talent	isSharedTalentOf
hasSpec	WoW_Class	Spec	isSpecOf
hasTalent	Spec	Talent	isTalentOf
hasTrait	Race	Trait	isTraitOf
hasWoW_Class	Race	WoW_Class	isWoW_ClassOf
hasZone	Location	Zone	isZoneOf

Tabela 2: Tabela de Object Properties

1.2.4 Modelo Físico

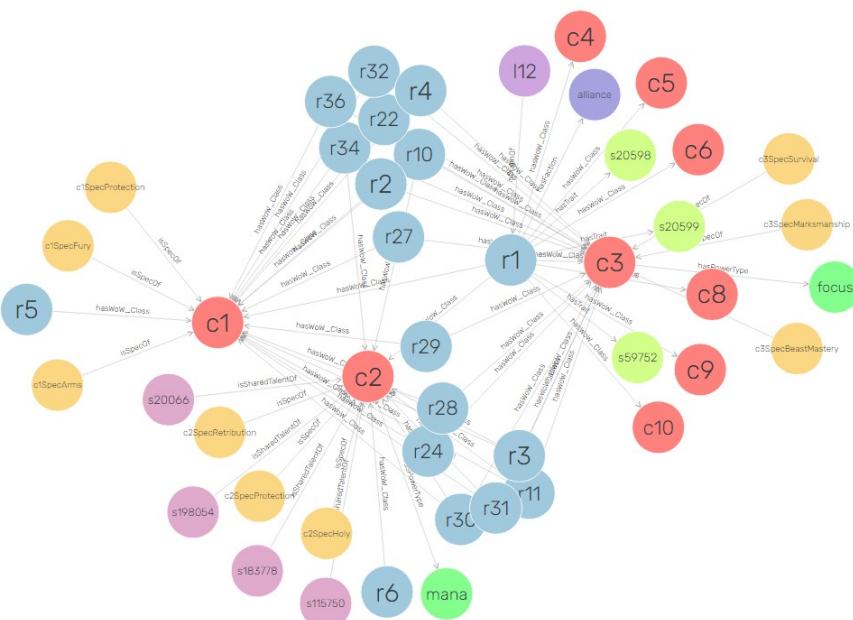


Figura 1: Exemplo do Grafo gerado por GraphDB



2 Implementação do Back-End

Na implementação do **Back-End**, foi criado de um servidor **REST**, desenvolvido em **nodeJS**, com o auxilio da framework **express**, capaz de responder a pedidos do cliente.

Para tal foi necessário implementar métodos de acesso à base de dados assim como, uma correta autenticação dos utilizadores.

2.1 Estrutura do servidor

```
→ wowREST git:(master) X tree
.
├── app.js
├── bin
│   └── www
├── controllers
│   └── wow.js
├── package-lock.json
├── package.json
└── routes
    ├── classes.js
    ├── factions.js
    ├── npcs.js
    ├── races.js
    ├── spells.js
    └── zones.js
```

Figura 2: Árvore da estrutura dos ficheiros do servidor

2.2 Rotas de acesso

Para o correto funcionamento do servidor, foi necessário implementar rotas que respondem, em **JSON** a diferentes pedidos.

URL: <http://localhost:7865/>

Método	Rota	SubRota	Descrição
GET	/factions	/	Lista de Frações
GET	/factions	/:f	Fração por id (:f)
GET	/factions	/:f/races	Raças pertencentes à fração f
GET	/factions	/:f/capitals	Localizações principais da fração
GET	/races	/	Lista de Raças
GET	/races	/:r	Raça por id (:r)

Tabela 3: Tabela de Rotas - parte 1



Método	Rota	SubRota	Descrição
GET	/races	/:r/traits	Lista de Traits da raça r
GET	/races	/:r/location	Localização inicial da raça r
GET	/races	/:r/classes	Classes disponíveis à raça r
GET	/classes	/	Lista de Classes
GET	/classes	/:c	Classe por id (:c)
GET	/classes	/:c/specs	Lista de Specs da class c
GET	/classes	/:c/talents	Lista de Talentos da class c
GET	/spells	/	Lista de Abilidades
GET	/spells	/mounts	Lista de Mounts
GET	/spells	/aquatic	Lista de mounts aquáticas
GET	/spells	/ground	Lista de mounts terrestres
GET	/spells	/flying	Lista de mounts voadoras
GET	/spells	/:s	Abilidade por id (:s)
GET	/zones	/	Lista de Zonas
GET	/zones	/raids	Lista de Raids
GET	/zones	/dungeons	Lista de Dungeons
GET	/zones	/:z	Zona por id (:z)
GET	/zones	/:z/bosses	Lista de Boss da zona z
GET	/npc	/:npc	

Tabela 4: Tabela de Rotas



3 Interface da Aplicação

3.1 Página Home

Página inicial da aplicação, com o nome da aplicação.

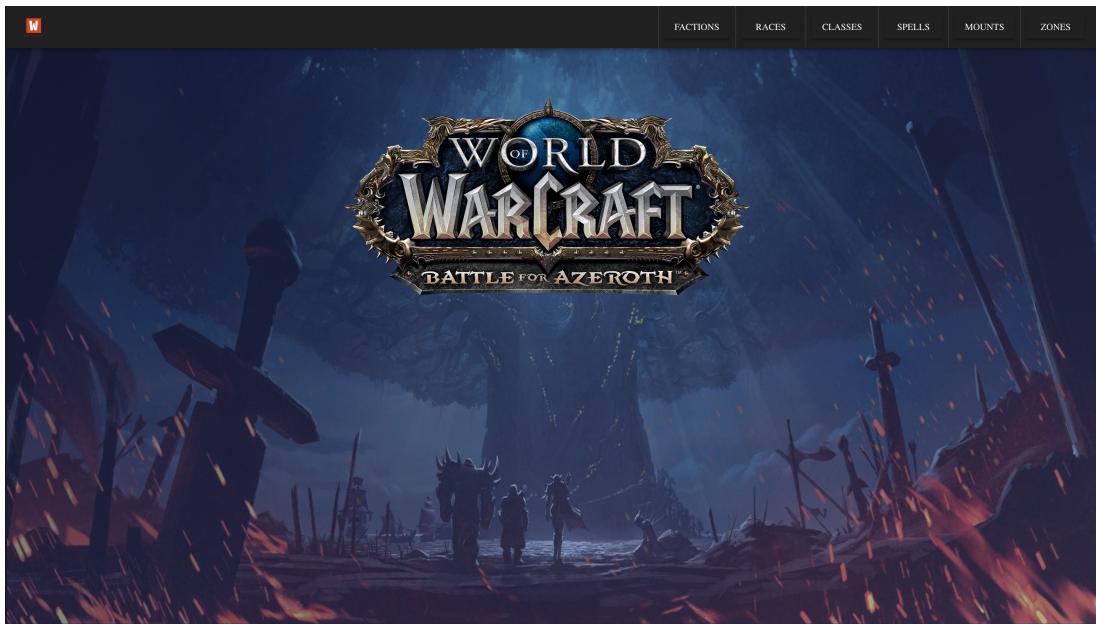


Figura 3: Página Home

3.2 Página de Fração

Nesta página estará apresentada uma fração assim como as cidades capitais.

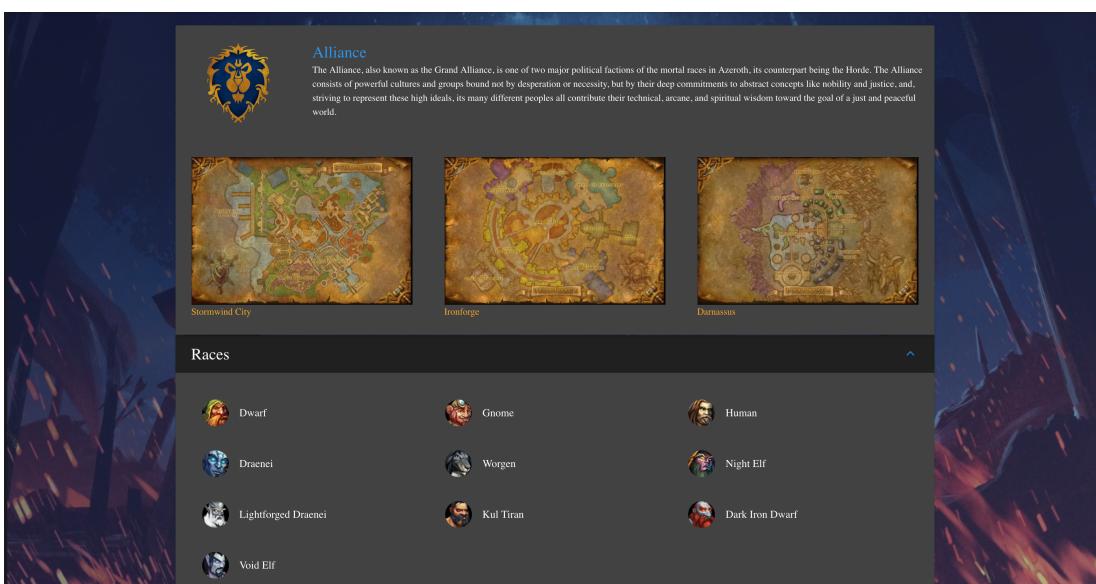


Figura 4: Página de Fração



3.3 Página de Raça

Nesta página estará apresentada uma raça e quais as classes disponíveis à mesma.

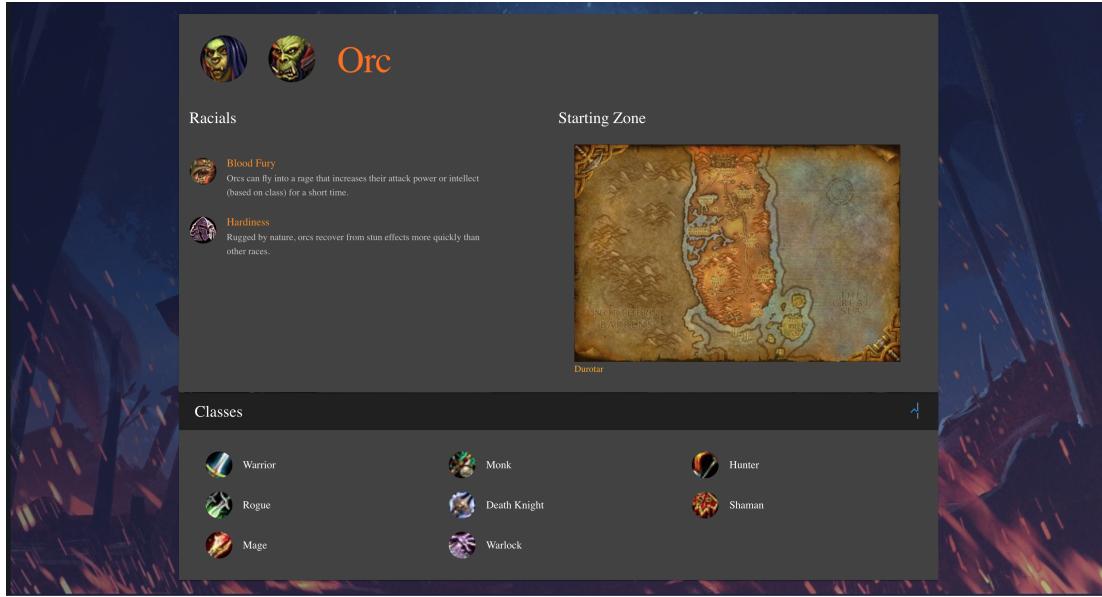


Figura 5: Página de Raça

3.4 Modal de Spell

Neste modal é apresentado um spell e toda a informação por ele pertencente.

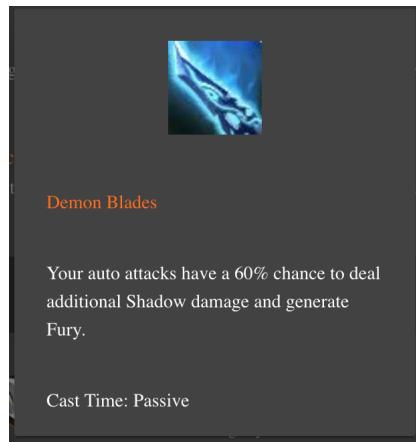


Figura 6: Modal de Spell



3.5 Página de Class

Nesta página é apresentado, com detalhe toda uma classe, os seus traços e talent trees.



Figura 7: Página de Class

3.6 Lista de Spells

Lista que contém todos os spells disponíveis na aplicação

Spells		
Icon	Name	Description
	A Murder of Crows	Summons a flock of crows to attack your target, dealing 1,136 Physical damage over 15 sec. If the target dies while under attack, A Murder of Crows' cooldown is reset.
	Aberration	Worgen have a natural resistance to Shadow and Nature discipline.
	Absolute Corruption	Corruption is now permanent and deals 15% increased damage. Duration reduced to 24 sec against players.
	Abundance	For each Rejuvenation you have active, Regrowth's cost is reduced by 6% and critical effect chance is increased by 6%.
	Abyssal Strike	Infernal Strike's range is increased by 10 yards, and its cooldown is reduced by 8 sec.
	Aerobic Strikes	Increases the range on all your melee attacks by 3 yards.
	Aegis of Light	Channels an Aegis of Light that protects you and all allies standing within 10 yards behind you for 6 sec, reducing all damage taken by 20%.
	Afterlife	Increases the duration of Spirit of Redemption by 50%.
	Aftershock	Earth Shock and Earthquake have a 25% chance to refund the Maelstrom spent on them.
	Agonizing Flames	Immolation Aura increases your movement speed by 30%, and deals 20% increased damage.

Figura 8: Lista de Spells



3.7 Lista de Zonas

Estão representadas duas listas, que dividem os tipos de zonas.

Raids				Dungeons			
Name	Location	Level Required	Number of Players	Name	Location	Level Required	
Ahn'Qiraj Temple	Silithus	60	40	Ahn'kahet: The Old Kingdom	Dragonflight	61	
Antorus, the Burning Throne	Antorus, the Burning Throne	110	10-30	Assault on Violet Hold	Violet Hold	105	
Baradin Hold	Tol Barad	85	10/25	Atal'Dazar	Atal'Dazar	120	
Battle of Dazar'alor	Dazar'alor	120	10-30	Auchenai Crypts	Terokkar Forest	63	
Black Temple	Shadowmoon Valley	70	30	Auchindoun	Tolador	94	
Blackrock Foundry	Gorgrond	100	10-30	Azjol-Nerub	Dragonflight	60	
Blackwing Descent	Blackrock Mountain	85	10/25	Black Rook Hold	Val'sharah	110	
Blackwing Lair	Blackrock Mountain	60	40	Blackfathom Deeps	Ashenvale	20	
Crucible of Storms	Crucible of Storms	120	10-30	Blackrock Caverns	Blackrock Mountain	80	
Dragon Soul	Dragon Soul	85	10/25	Blackrock Depths	Blackrock Mountain	47	

Figura 9: Lista de Zonas

3.8 Página de Zonas

Página que demonstra uma zona e os seus bosses

The screenshot shows a detailed page for the Blackrock Foundry zone. At the top, there's a navigation bar with tabs for FACTIONS, RACES, CLASSES, SPELLS, MOUNTS, and ZONES. Below the navigation bar, the title "Blackrock Foundry" is displayed in large green text. A descriptive paragraph follows, mentioning the foundry's history as the ancestral home of the Blackrock orcs and its current role as the center of the Iron Horde's military might. It specifies a minimum level of 100 and a maximum level of 100. The number of players required is listed as 10-30, and the number of floors is 5. To the right of the text, there's a large, detailed map of the zone, showing various locations like "IRON SHREWDREEF" and "GORGROND". Below the map, a section titled "Bosses" lists the bosses of the zone. The background of the page features a dark, fiery landscape with molten lava and industrial structures.

Figura 10: Página de Zona



3.9 Modal de Boss

Este modal representa os detalhes de um boss.

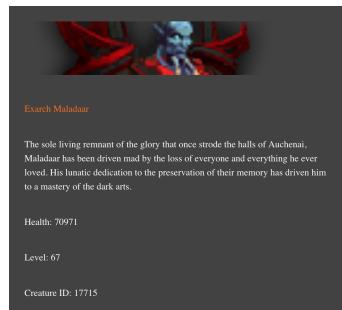


Figura 11: Modal de Boss



Conclusão

Com a análise dos dados, foi necessária a utilização de *scripts* de forma a converter os dados para **turtle**, além disso, foi possível filtrar a informação necessária para a tornar, desta maneira o mais coerente possível.

Após a análise dos dados, é feito o desenho da ontologia. Nesta são verificadas as classes pretendidas, os seus atributos e as relações entre elas.

Com a ontologia devidamente especificada, foram idealizadas queries, em **sparql**, capazes de devolver dados de forma predefinida, e desta maneira garantir o funcionamento da aplicação. De seguida, foi criado o servidor **REST**, como ponte de comunicação entre cliente e a base de dados, **GraphDB**, onde este trata a estrutura recebida e normaliza os seus dados.

Por fim, a aplicação reativa, escrita em **Vue**, mostra os dados aos clientes, e para isso, foram desenhadas várias páginas, onde cada uma trata uma respetiva parte de informação. De forma a embelezar a aplicação foi utilizado **vuetify**, onde o tratamento de grelhas, estruturas de listas e tabelas, predominam na aplicação.

Como trabalho futuro, é possível "escalar"em termos de funcionalidades esta aplicação, dispondo para isso, de novas funcionalidades.