

Projeto de Bases de Dados (CC2005)

Universo de *Game of Thrones*

Grupo nº G3A

Nº mecanográfico	Nome
202207859	Diogo Alberto Manin Rei Vila
202300247	Sofia Oliveira Bucci
202200336	Sophia Cheto de Queiroz Fonseca
202300916	Yan de Oliveira Christiano Coelho

1. Universo considerado

Tema: Game of Thrones

O universo considerado para a criação da base de dados (BD) consiste em informações sobre o mundo fictício da série *Game of Thrones*. De forma resumida, a narrativa principal gira em torno da disputa pelo trono após a morte do rei, que não deixou herdeiros legítimos. Essa situação desencadeia uma série de batalhas entre diferentes personagens que reivindicam o direito ao trono.

A base de dados, portanto, concentra-se nesses conflitos, registrando informações como as batalhas ocorridas, as famílias aliadas, os personagens que lideram os ataques ou defesas, os locais onde os confrontos acontecem, entre outros detalhes relevantes que ocorrem dentro do continente de Westeros (local que é o foco da série). Com isso, foi possível organizar a seguinte estrutura para representar esse universo:

- O continente é dividido em 10 regiões (Region);
- Cada região possui uma Família/Casa (House) que a governa, *exceto a região "Beyond the Wall", que não possui governo estabelecido, pois é uma região fora da civilização;*
- Cada região tem diversas Casas espalhadas pelo seu território;
- Cada casa está localizada numa Cidade (City) de uma certa região;
- Cada Casa/Família possui diversos personagens (Character) que a compõem.

Batalhas (Battle):

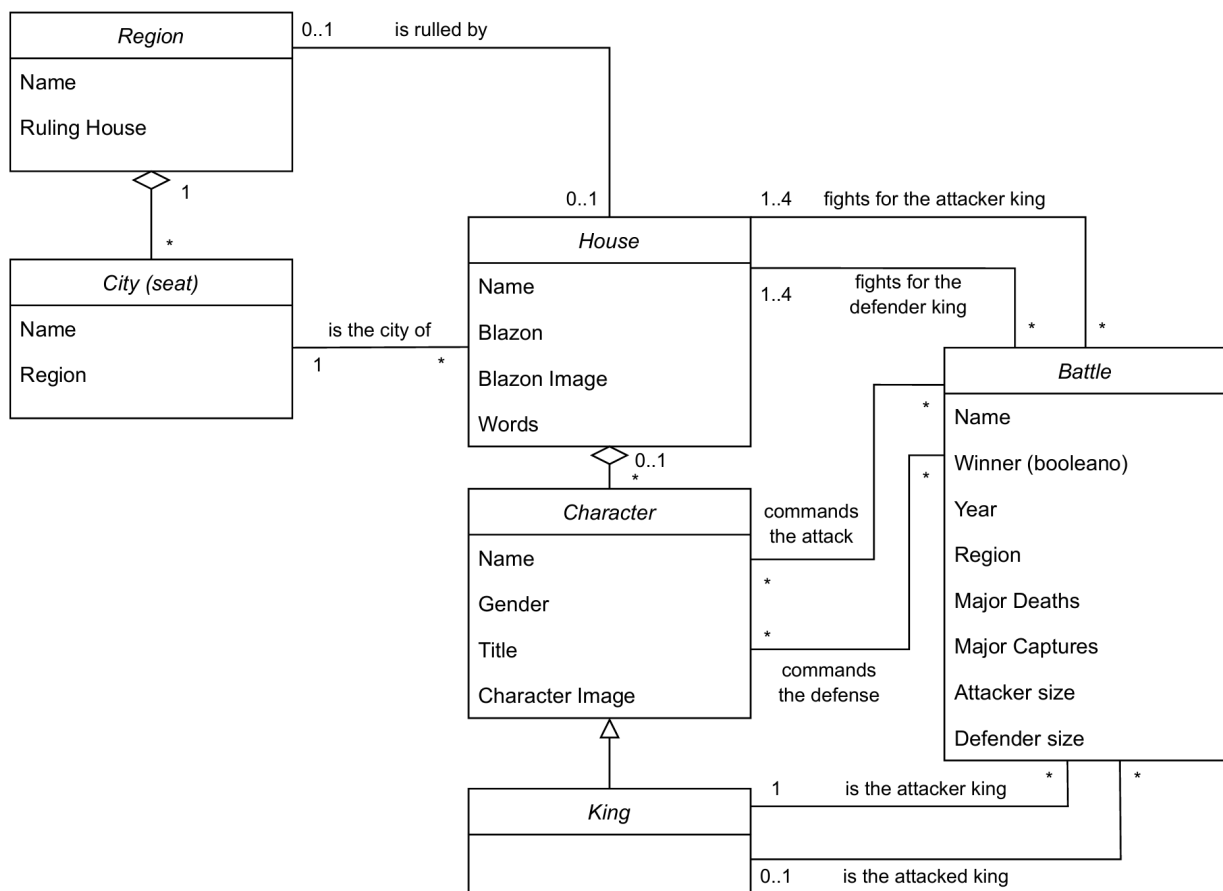
- Cada batalha possui um rei que ordena o ataque, porém nem sempre a defesa (grupo atacado) possui um rei na sua liderança;
- Cada batalha possui ao menos um comandante no ataque, porém nem sempre há algum na defesa;
- Cada lado da batalha possui ataques e defesas de diversas famílias que estão aliadas ao rei do seu respectivo lado;

- Além disso, existem informações estatísticas associadas a cada batalha, como o tamanho de cada exército, se houve mortes de personagens importantes, etc

Utilizamos, para isso, três conjuntos de dados, referenciados no fim deste relatório:

Conjunto de dados	Nº de entradas
Battles	10
Characters	223
Houses	448

2. Modelo de classes UML

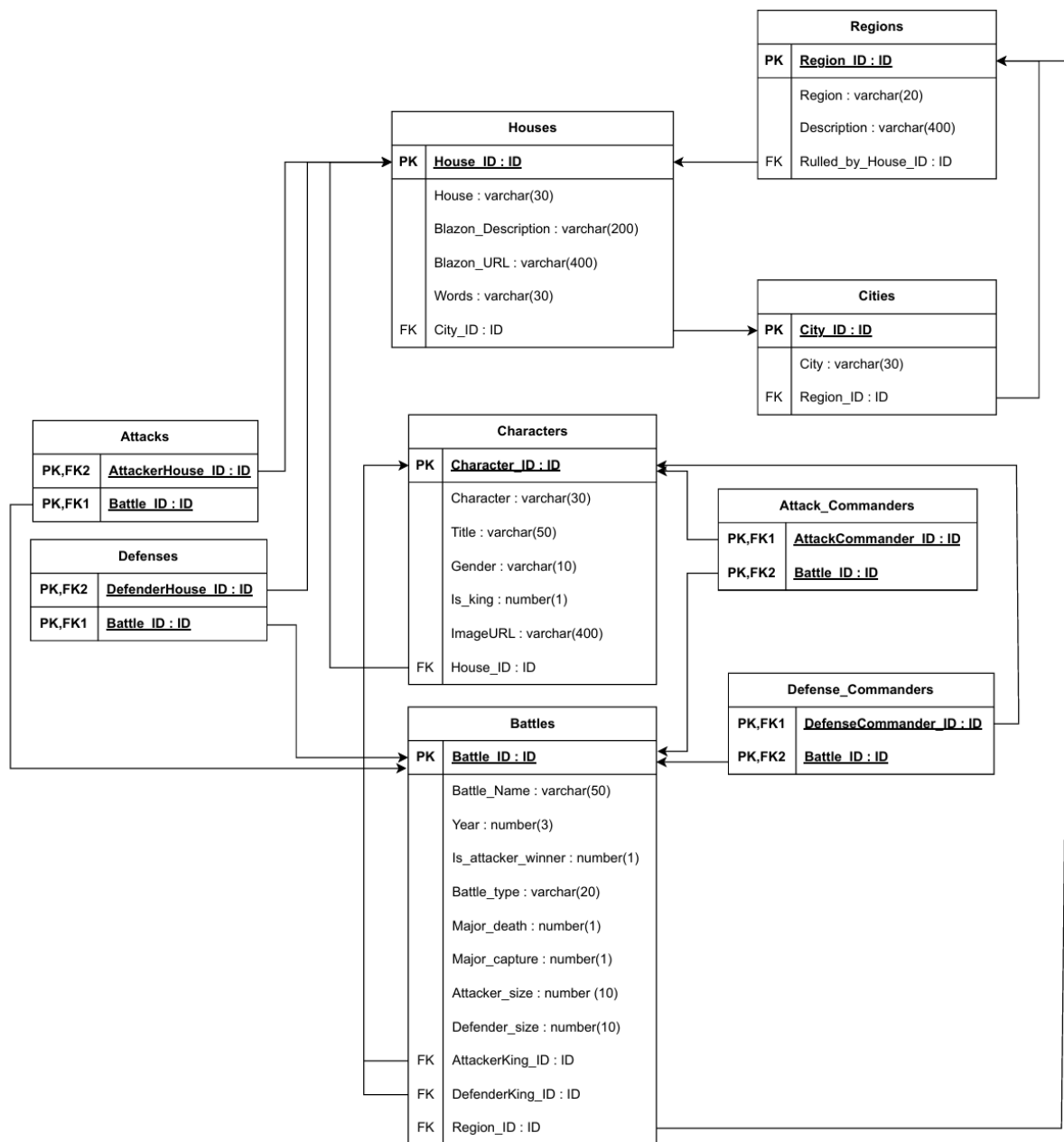


Analisando os dados sobre as batalhas, percebemos que existiam no máximo 4 comandantes por batalha. Assim, tínhamos duas opções para representá-los:

1. Adicionar na classe Battle 4 atributos, um para cada comandante, que poderiam ser nulos quando não houvesse todos os 4
2. Cada comando ser representado como uma associação entre o Character e a Battle

Decidimos representá-los da segunda forma, uma vez que a maioria das batalhas tinham no máximo 2 comandantes. Assim, conseguimos reduzir o espaço que seria utilizado desnecessariamente na base de dados com tantos espaços a nulo.

3. Modelo relacional



Para traduzirmos as tabelas *Attacks* e *Defenses* para o modelo relacional, optamos por criar uma tabela (para cada) com duas chaves primárias, uma como chave externa para a tabela *Battles* (Battle_ID) e outra como chave externa para a tabela *Houses* (House_ID). Fizemos assim para evitar o crescimento desnecessário de atributos da tabela *Battles*, visto que algumas batalhas possuem até quatro “atacantes” e teríamos que criar quatro novos atributos na tabela *Battles* como alternativa. Além disso, é uma solução mais escalável, facilitando a expansão caso necessário.

3. Povoamento de tabelas

Para a obtenção de dados, utilizamos 3 tabelas de diferentes fontes (citadas com links nas “referências”), das quais obtivemos as seguintes colunas:

1. Tabela de Personagens:
 - a. Nome
 - b. Título
 - c. Família/Casa
 - d. URL da Imagem
2. Tabela de Casas:
 - a. Região
 - b. Casa
 - c. URL do brasão
 - d. Descrição do brasão
 - e. Cidade
 - f. Slogan/Lema
3. Tabela de Batalhas:
 - a. Nome
 - b. Ano
 - c. Rei ordenou o ataque
 - d. Rei que foi atacado
 - e. Casas que lutaram no ataque
 - f. Casas que lutaram na defesa
 - g. Resultado da batalha (se o ataque foi vencedor)
 - h. Quantidade de mortes importantes
 - i. Quantidade de capturas de personagens importantes
 - j. Tamanho da defesa
 - k. Tamanho do ataque
 - l. Comandante da defesa
 - m. Comandante do ataque
 - n. Região em que ocorreu

De acordo com os dados disponíveis, fizemos pequenas alterações manuais nos valores de alguns campos, como nome da casa ou nome do personagem, para que todas as ocorrências tivessem compatibilidade de escrita, maiúsculas etc.

Nos campos que possuíam mais de um valor numa só célula (separados por vírgulas), foram criadas novas colunas usando a função SPLIT do Google Sheets.

Além disso, a coluna “is_king”, que surgiu da classe “King” do modelo UML, foi obtida a partir dos personagens que estão presentes na tabela das batalhas como “defender king” ou “attacker king”.

Após termos todas as colunas bem formatadas em cada tabela, criamos uma nova tabela para cada classe, utilizando as seguintes ferramentas:

- Filtro de dados com repetição
- Geração de índices automática
- Função PROCV para substituir os valores de uma tabela pelo seu ID

Nome da tabela	Nº de entradas
Regions	10
Cities	223
Houses	448
Characters	85
Battles	36
Attacks	50
Defenses	38
Attack Commanders	53
Defense Commanders	38

4. Interrogações SQL

NAVEGAÇÃO ENTRE PÁGINAS:

As queries simples para listagem de elementos da tabela ou de seleção de um elemento específico seguiram o padrão seguinte:

```
SELECT {colunas sem ID}
FROM {tabela} JOIN {tabela}
WHERE {id} = ?
```

Além disso, utilizamos algumas queries mais específicas para mostrar mais informações relacionadas a um elemento, que aparecem na sua página individual

1. Casa governante de uma região

```
SELECT House_ID, House
FROM Houses JOIN Regions
ON House_ID = Ruled_by_House_ID
WHERE Region_ID = ?
```

2. Batalhas em que uma determinada casa lutou

```
SELECT battle_name, battle_ID, attackerhouse_id
FROM battles natural JOIN attacks
WHERE attackerhouse_id=?
UNION
SELECT battle_name, battle_ID, defenderhouse_id
FROM battles natural JOIN defenses
WHERE defenderhouse_id=?
```

3. Batalhas que um determinado personagem comandou

```
SELECT battle_id, battle_name
FROM attack_commanders natural JOIN battles
WHERE attackcommander_id=?
UNION
SELECT battle_id, battle_name
FROM defense_commanders natural JOIN battles
WHERE defense_commander_id=?
```

4. Pesquisa + código Python para evitar SQL Injection

Considerando que todos os valores dos elementos pesquisados são compostos apenas por letras, usamos a função *isalpha* para garantir que nenhum caractere específico de utilização em queries fosse passado para interação com a base de dados

```
if (not str.isalpha(expr)):
    abort(404, 'expression is not valid')

search = { 'expr': expr }
expr = '%' + expr + '%'
houses = db.execute(
    '''
    SELECT House, House_ID, Words
    FROM Houses
    WHERE House LIKE ?
    ''', [expr]).fetchall()
```

CURIOSIDADES

1. Casas mais atacadas:

Quais casas foram mais frequentemente atacadas e quantos foram esses ataques?

```
SELECT
    h.House AS Defender_House,
    COUNT(d.Battle_ID) AS Attacks
FROM
    Houses h
JOIN
    Defenses d
ON h.House_ID = d.DefenderHouse_ID
GROUP BY
    h.House
ORDER BY
    Attacks DESC;
```

	Defender_House	Attacks
1	House Lannister	9
2	House Stark of Winterfell	8
3	House Baratheon of King's Landing	5
4	House Tully of Riverrun	4
5	House Greyjoy of Pyke	3
6	House Tyrell of Highgarden	2
7	Night's Watch	1
8	House Mallister of Seagard	1
9	House Frey of Riverrun	1
10	House Darry of Darry	1

2. Batalhas com maior tropa envolvida:

Quais foram as batalhas com o maior número de pessoas que lutaram tanto no ataque quanto na defesa?

```
SELECT
    Battle_Name AS Battle,
    (Attacker_size +
    Defender_size) AS Total_Tropas
FROM
    Battles
WHERE
    Attacker_size IS NOT NULL
AND
    Defender_size IS NOT NULL
ORDER BY
    Total_Tropas DESC;
```

	Battle	Total_Tropas
1	Battle of Castle Black	101240
2	Battle of the Green Fork	38000
3	Battle of the Fords	30000
4	Battle of the Blackwater	28250
5	Battle of Riverrun	25000
6	Siege of Storm's End	25000
7	Battle of the Golden Tooth	19000
8	Battle of the Camps	18625
9	Battle of Oxcross	16000
10	Siege of Winterfell	13000

3. Regiões com mais batalhas:

Quais regiões de Westeros tiveram o maior número de batalhas?

```
SELECT
    r.Region AS Region,
    COUNT(b.Battle_ID) AS Qtd_Battles
FROM
    Regions r
JOIN
    Battles b ON r.Region_ID =
    b.Region_ID
GROUP BY
    r.Region
ORDER BY
    Qtd_Battles DESC;
```

	Region	Qtd_Battles
1	Riverlands	15
2	North	10
3	Westerlands	3
4	Stormlands	3
5	Reach	2
6	Crownlands	2
7	Beyond the Wall	1

4. Batalhas mais equilibradas:

Quais batalhas tiveram as forças mais equilibradas entre os atacantes e defensores? Ou seja, onde a diferença entre os atacantes e defensores foi a menor.

```
SELECT
    Battle_Name,
    ABS(Attacker_size -
        Defender_size) AS Size_Diff
FROM
    Battles
WHERE
    Attacker_size IS NOT NULL AND
    Defender_size IS NOT NULL
ORDER BY
    Size_Diff ASC
```

	Battle_Name	Size_Diff
1	Sack of Harrenhal	0
2	The Red Wedding	0
3	Battle of Torrhen's Square	656
4	Sack of Winterfell	1382
5	Battle of the Green Fork	2000
6	Siege of Winterfell	3000
7	Battle of Oxcross	4000

5. Regiões com mais casas:

Quais regiões de Westeros possuem o maior número de casas?

```
SELECT
    r.Region AS Region_Name,
    COUNT(h.House_ID) AS Total_Houses
FROM
    Regions r
JOIN
    Cities c ON r.Region_ID = c.Region_ID
JOIN
    Houses h ON c.City_ID = h.City_ID
GROUP BY
    r.Region
ORDER BY
    Total_Houses DESC;
```

	Region_Name	Total_Houses
1	Reach	72
2	North	69
3	Riverlands	57
4	Westerlands	53
5	Crownlands	50
6	Iron Islands	40
7	Stormlands	39

6. Número médio de casas por região:

Qual é o número médio, máximo e mínimo de casas por região em Westeros?

```
SELECT AVG(house_count) AS avg,  
MAX(house_count) AS max,  
MIN(house_count) AS min  
FROM (  
  SELECT Region_ID, COUNT(*) AS house_count  
  FROM Cities c  
  JOIN Houses h ON c.City_ID = h.City_ID  
  GROUP BY Region_ID)
```

	avg	max	min
1	44.8	72	1

7. Taxa de vitória em ataques de cada rei (pretendente ao trono)

Qual é a porcentagem de vitórias em batalhas de cada rei atacante?

```
SELECT  
  c.Character_ID as king_ID,  
  c.Character AS Attacker_King,  
  COUNT(b.Battle_ID) AS Total_Battles,  
  SUM(b.is_attacker_winner) AS Total_Wins,  
  ROUND((SUM(b.is_attacker_winner) * 100.0) / COUNT(b.Battle_ID), 2)  
AS WinRate  
FROM Characters c JOIN Battles b  
ON c.Character_ID = b.AttackerKing_ID  
GROUP BY c.Character  
ORDER BY WinRate DESC;
```

	king_ID	Attacker_King	Total_Battles	Total_Wins	WinRate
1	153	Balon Greyjoy	7	7	100
2	113	Joffrey Baratheon	14	13	92.86
3	111	Robb Stark	10	8	80
4	118	Stannis Baratheon	5	2	40

8. Percentagem de personagens por gênero:

Qual é a distribuição percentual de personagens por gênero em Westeros?

```
SELECT
    Gender,
    (COUNT() * 100.0 /
     (SELECT COUNT() FROM Characters))
    AS percentage
FROM Characters
GROUP BY Gender;
```

	Gender	percentage
1	Female	20
2	Male	80

5. Aplicação Python

"Endpoint"	Funcionalidade
/	Página de entrada do site.
/regions	Página que mostra o mapa com a lista das regiões e os brasões da sua casa governante no seu respectivo local do mapa.
/regions/<int:id>/	Página de cada região específica.
/houses	Página com lista das casas.
/houses/<int:id>/	Página de cada casa específica.
/battles	Página com lista das batalhas.
/battles/<int:id>/	Página de cada batalha específica.
/cities	Página com lista das cidades.
/cities/<int:id>/	Página de cada cidade específica.
/characters	Página com lista dos personagens.
/characters/<int:id>/	Página de cada personagem específico.
/region/region_id/houses	Página da lista de casas na região específica.
/region/region_id/cities	Página da lista de cidades na região específica.

/region/region_id/battles	Página da lista de batalhas na região específica.
/houses/search/<expr>/	Pesquisa casas com os caracteres digitados na busca.
/battles/search/<expr>/	Pesquisa batalhas com os caracteres digitados na busca.
/cities/search/<expr>/	Pesquisa cidades com os caracteres digitados na busca.
/characters/search/<expr>/	Pesquisa personagens com os caracteres digitados na busca.
/curiosities	Página que exhibe algumas curiosidades sobre o universo considerado.

6. Conclusão

Este projeto nos permitiu a exploração detalhada de todo o universo de *Game of Thrones*, incluindo as inter-relações entre regiões, casas e personagens, além de eventos de batalhas importantes para a construção da narrativa da série. A criação de um modelo relacional bem definido, com mapeamento de classes UML e conversão para tabelas em modelo relacional, desenvolveu a nossa capacidade de traduzir um cenário complexo como o universo de *Game of Thrones* em um sistema de banco de dados funcional e coerente.

O processo de povoamento de dados foi um tanto desafiador, visto que o fizemos utilizando diferentes fontes (como listado nas referências). Assim, tivemos que aplicar ferramentas para refinar as informações e garantir consistência. Dentre os desafios, ficam destacados a padronização de nomenclaturas e o tratamento de valores multivalorados, que foram superados por meio de ferramentas específicas, como a função *SPLIT* e a criação de índices automáticos, como já citado na seção Povoamento.

A implementação dos endpoints em Python nos mostrou ser uma abordagem prática e eficiente para a integração de dados com aplicações web, facilitando o acesso a informações relevantes de forma dinâmica e organizada. Isso foi complementado por combinações de consultas SQL para manipular dados e extrair informações do banco de dados de acordo com o que achamos relevante.

No fim, o projeto combinou modelagem de dados, desenvolvimento de aplicações e o uso de fontes variadas, resultando em uma base de dados que não apenas documenta o universo de *Game of Thrones*, mas também serve como ferramenta prática para explorar seus pormenores. Assim, apesar de termos escolhido um tema diferente e mais desafiador para a realização deste trabalho, conseguimos colocar em prática o que nos foi ensinado na cadeira, sendo o projeto crucial para a consolidação do que aprendemos durante o semestre.

Referências

Datasets

1. Tabela de Personagens:
<https://www.kaggle.com/datasets/ankytastic/game-of-thrones-characters>
2. Tabela de Batalhas:
<https://www.kaggle.com/datasets/satyampandey4229/games-of-thrones>
3. Tabela de Casas e Regiões:
<https://www.kaggle.com/datasets/neelgajare/463-game-of-thrones-houses-dataset>

Imagens e Pesquisa de Informações

1. <https://thronesapi.com>
2. <https://gameofthrones.fandom.com/wiki/Wiki>
3. https://awoiaf.westeros.org/index.php/Main_Page