



UNIVERSIDADE DE SÃO PAULO  
ESCOLA DE ENGENHARIA DE SÃO CARLOS  
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO

**Bases de Dados - SCC0640**

# **INTERGALACTIC EMPIRES**

**Artur Brenner Weber - 12675451**

**Guilherme Castanon Silva Pereira - 11801140**

**Gustavo Moura Scarenci de Carvalho Ferreira - 12547792**

**Matheus Henrique Dias Cirillo - 12547750**

**Docente responsável:** Prof.<sup>a</sup> Elaine Parros Machado de Sousa

São Carlos  
2º semestre/2023

# SUMÁRIO

<b>1</b>	<b>Descrição do problema e dos requisitos de dados</b>	<b>1</b>
1.1	Descrição do sistema	1
1.1.1	Turnos	1
1.1.2	Histórico	1
1.1.3	Império	1
1.1.4	Planeta	2
1.1.5	Colônia	3
1.1.6	Batalha	3
1.1.7	Exploração	4
1.1.8	Recursos	5
1.1.9	Prioridade	5
1.1.10	Estoque	5
1.1.11	Espécie	5
1.1.12	Povos	5
1.1.13	Civil	7
1.1.14	Cientista	7
1.1.15	Militar	7
1.1.16	Estruturas	8
1.1.17	Instituição de Ensino	8
1.1.18	Indústria	8
1.1.19	Base Militar	8
1.1.20	Atuação	8
1.1.21	Construção	9
1.1.22	Tecnologia	9
1.1.23	Manufatura	10
1.1.24	Armamento	11
1.1.25	Transporte	11
1.1.26	Movimento de estoque e povo	11
1.2	Estratégia Geral	11
1.2.1	Fase de Preparativos (Primeira Fase)	11
1.2.2	Fase de Cooperação Imperial (Segunda Fase)	11
1.2.3	Fase de Batalha (Terceira Fase)	12
1.2.4	Colonização e Domínio:	12
1.2.5	Desenvolvimento Tecnológico	12
1.2.6	Diplomacia e Conflito	12
1.2.7	Gerenciamento de Recursos	12
1.3	Notas do Modelo	13
1.4	Principais operações	15
1.4.1	Império	15
1.4.2	Consultas	15

<b>2</b>	<b>Modelo Entidade-Relacionamento</b>	<b>16</b>
<b>3</b>	<b>Mudanças realizadas em relação à primeira entrega</b>	<b>17</b>
3.1	Na descrição do Problema e dos Requisitos de Dados . . . . .	17
3.2	No MER-X . . . . .	17
<b>4</b>	<b>Modelo Relacional</b>	<b>19</b>
4.1	Modelo . . . . .	19
4.2	Justificativas dos Mapeamentos . . . . .	20
4.2.1	Império . . . . .	20
4.2.2	Planeta . . . . .	20
4.2.3	Prioridade de produção . . . . .	20
4.2.4	Povo . . . . .	20
4.2.5	Conhecimento . . . . .	20
4.2.6	Estrutura . . . . .	21
4.2.7	Tecnologia . . . . .	21
4.2.8	Estoque . . . . .	21
4.2.9	Construção . . . . .	21
4.2.10	Colônia . . . . .	22
4.2.11	Exploração . . . . .	22
4.2.12	Batalha . . . . .	22
4.2.13	Estoque_gasto_batalha . . . . .	22
4.2.14	povo_morto_batalha . . . . .	22
4.2.15	Atuação . . . . .	23
4.2.16	Gera_recurso . . . . .	23
4.2.17	move_recurso . . . . .	23
4.2.18	move_povo . . . . .	23
<b>5</b>	<b>Mudanças realizadas em relação à segunda entrega</b>	<b>25</b>
5.1	No relatório . . . . .	25
5.2	No DER . . . . .	25
5.3	No MR . . . . .	25
<b>6</b>	<b>Consultas do Projeto</b>	<b>26</b>
6.1	Quantidade de Recursos de dado Império em um dado Turno . . . . .	26
6.2	Quantidade de Recursos Produzidos por um dado Império até Dado Turno . . . . .	26
6.3	Quantidade de Civis Especializados de cada Espécie por Determinado Império até Dado Turno . . . . .	27
6.4	Quantidade de Indivíduos de Cada Espécie Mortos em Batalhas Protagonizadas por dado Império . . . . .	28
6.5	Chechagem se Dado Império Consegue Construir Dada Estrutura com seu Estoque Disponível em dado Turno . . . . .	28
6.6	Listagem de planetas que já construíram todas as estruturas avançadas . . . . .	29
6.7	Classificação dos Impérios por Estimativa de Potencial de Poder Total em Dado Turno	30
6.8	Estimativa de Potencial de Poder de Cada Colônia de cada Império em um dado Turno	31

<b>7</b>	<b>Descrição da Implementação do Projeto</b>	<b>32</b>
7.1	Organização e Arquitetura do Sistema . . . . .	32
7.2	Explicação sobre a Aplicação . . . . .	32
7.2.1	Visão Geral . . . . .	32
7.2.2	Detalhes técnicos da implementação . . . . .	33
7.2.3	Controle Transacional do Código . . . . .	33
7.2.4	Comandos SQL utilizados na aplicação . . . . .	34
7.2.5	Para rodar o projeto . . . . .	35
<b>8</b>	<b>Conclusão</b>	<b>37</b>
<b>9</b>	<b>Referências</b>	<b>38</b>

## 1. DESCRIÇÃO DO PROBLEMA E DOS REQUISITOS DE DADOS

### 1.1. Descrição do sistema

Intergalactic Empires é um jogo de turnos onde cada jogador assume o papel de um líder imperial intergaláctico visando expandir seu **império**. O objetivo do sistema criado é modelar como os diversos fatores que envolveriam relações interplanetárias se afetariam. Além disso, é fundamental que o projeto promova, como um dinâmico Turn-Based Strategy Game, a interação entre jogadores. Isso é feito via mecanismos que exijam conflito para crescimento dos usuários, evitando que o jogo fique parado por muitos turnos. Outro propósito do videogame é ser razoavelmente fidedigno à realidade no que tange as **estruturas** de um **planeta**, tentando ao máximo estabelecê-las para diversificar o comportamento de cada entidade. A exemplo, algumas estabelecem poder, algumas são consumíveis e outras realizam ações, aumentando a complexidade do gerenciamento do **império**. O usuário-alvo é qualquer pessoa que queira se divertir, interagir com outros e treinar seu cérebro estrategicamente.

Dentro do jogo, a meta é alcançar dominação total do universo, para isso eficientemente gerenciando as **colônias** de seu **império**, cada qual auxiliando o objetivo com suas próprias aptidões, de maneira eficiente. O jogo acaba quando 100% dos **planetas** forem *colonizados* por um **império**.

#### 1.1.1 Turnos

O jogo é baseado em turnos, onde cada turno é composto por ações de todos os jogadores e é faseado. Na primeira fase os jogadores devem fazer preparativos focados em evolução de **colônias**. Nesse momento cada usuário pode realizar 3 ações de gerenciamento colonial (dentre recrutamento, estudo, trabalho e construção) em cada colônia. Na segunda fase os jogadores têm direito a 3 ações fixas mais 1 ação a cada 3 colônias, focando em cooperação imperial (movimentação de recursos e povos e exploração de planetas). Por fim, na terceira fase os impérios entram em batalha, com cada jogador podendo realizar 2 ataques em outros planetas, mais 1 ataque a cada 5 colônias. Quando o último jogador faz sua última ação de batalha, se encerra um turno e o jogo calcula o resultado das ações. Turnos representam o tempo discretizado da progressão do jogo e são peça essencial no armazenamento de histórico.

#### 1.1.2 Histórico

Ao final do jogo, é interessante que os jogadores possam ver sua evolução, ao estilo Age of Empires II. Logo, é de extrema importância salvar certos aspectos específicos do estado do jogo ao final de cada turno. Estes incluem **explorações, construções, colônias, estoques, povos, batalhas, construções e prioridade**.

#### 1.1.3 Império

Cada **império** é controlado por um jogador e é identificado pelo seu nome. Além disso, um império sabe a quantidade de **planetas** colonizados por ele e também tem uma cor capaz de identificá-lo. Para estender seu poder, **império** pode tentar *colonizar* um **planeta**, e, caso consiga, esse passará a ser sua **colônia**. Um império pode *explorar* qualquer **planeta**, desde que consiga alcançá-lo com a **tecnologia de transporte** que tem à disposição na **colônia** de origem da análise. Com as mesmas restrições, pode também tentar *colonizá-lo*. Para tal, o **império** precisa escolher

uma de suas **colônias** para *batalhar* com o **planeta** visado (que pode ou não ser **colônia** de algum outro jogador).

#### 1.1.4 Planeta

Cada um possui atmosfera, coordenadas únicas, população máxima, temperatura, massa, máxima quantidade de estruturas, gravidade, quantidade de água e fertilidade. Fertilidade se refere à suscetibilidade de gerar **recursos**. Além disso, **planeta** é identificado pelo seu nome e guarda o cálculo do seu poderio militar.

O poderio militar atrelado a um **planeta** é de suma importância na definição de uma determinada **batalha** envolvendo-o. Poderio militar é um valor numérico calculado através da quantidade de **militares** e suas agressividades, **armamentos** e seu nível, quantidade e eficiência das **bases militares** presentes no **planeta**. Ele é calculado através da seguinte expressão:

$$PM = \sum_{i=0}^n (Q_{Mi} \cdot A_i) + N_A! \cdot Q_A + 10 \cdot \sum_{j=0}^m (Q_{BMj} \cdot E_{BMj}) \quad (1.1)$$

- $PM$  = poderio militar;
- $n$  = número de **espécies** diferentes no **planeta**;
- $Q_{Mi}$  = número de **militares** de uma **espécie**  $i$  do **planeta**;
- $A_i$  = agressividade da **espécie**  $i$ ;
- $N_A$  = nível de **armamento**;
- $Q_A$  = quantidade de armas possuídas pelo **planeta**;
- $m$  = número de **bases militares** diferentes no **planeta**.
- $Q_{BMj}$  = quantidade de **bases militares**  $j$ ;
- $E_{BMj}$  = eficiência da **base militar**  $j$ .

Exemplo: Em marte, há 700 **militares** “Uspianos” com agressividade 0,9 e 500 **militares** “Federais” com agressividade 0,5. Além disso, o **planeta** é equipado com 1000 **armamentos** nível 2, além de 10 **bases militares** “ICMC” com eficiência 0,9 e 15 **bases militares** “EESC” com eficiência 0,2. Temos que o cálculo do índice poderio militar em marte se dará por:

$$PM = 700 \cdot 0,9 + 500 \cdot 0,5 + 2! \cdot 1000 + 10 \cdot (10 \cdot 0,9 + 15 \cdot 0,2) = 3000 \quad (1.2)$$

Dessa forma, o poderio militar de marte é 3000.

**Planetas** são *explorados* por **colônias** e também podem se tornar **colônias** do **império** ao serem *colonizados* em **batalhas**. Além disso, *possuem* uma quantidade de **recursos**, *contém* uma quantidade de **estruturas** e são *populados* por uma quantidade de **povos**. Ao participar de uma **batalha**, pode ser o *vencedor* dela. Além disso, em todo turno *prioriza* a produção de um **recurso**.

### 1.1.5 Colônia

Ao ser *colonizado* por um **império**, um **planeta** vira uma **colônia**, e a partir desse momento nunca deixa de ser uma, passando apenas pelas eventuais trocas de **império** dominante durante **batalhas**. Têm sempre um turno inicial e final em que pertencem ao **império**, sendo o final desconhecido enquanto continua de posse do mesmo jogador. Quando acontece uma **batalha** e o **império** dominante é trocado, o novo **império** herda todos os **recursos**, **estruturas** e **povos** remanescentes da **batalha**, aumentando sua área de dominação e contribuindo com seus objetivos principais. Uma **colônia** pode **batalha** com **planetas** no objetivo de dominá-los. É sempre possível identificar determinada **colônia** pelo turno em que foi dominada pelo **império**, o **império** e **planeta**.

### 1.1.6 Batalha

**Batalhas** são o método de expansão dos **impérios**. Para adquirir um novo **planeta**, é necessário lutar contra ele, não importando se é ou não colonizado. A vitória depende da existência de um poderio militar superior da **colônia** de ataque em relação ao **planeta** visado. Toda **batalha** ocorre em um turno e possui um **planeta vencedor**, um fator de violência da batalha e o custo de populações e **recursos**.

O vencedor é o atacante caso  $\frac{PM_a}{PM_d \cdot f_d} > 1$ , e, caso esse valor seja  $\leq 1$  o vencedor é o defensor. A violência da **batalha** é um valor entre 0 e 2 que altera os custos de **batalha**, dado pela média ponderada do fator de agressividade de todas as **espécies** envolvidas mais uma aleatoriedade entre 0 e 1 para considerar fatores inesperados. O perdedor sempre perde todos os **militares**, todas as armas,  $Q_N \cdot V_B$  naves e  $Q_G \cdot V_B$  combustíveis. Para cada **povo** do **planeta** perdedor morrem  $V_B \cdot \frac{PM_v}{PM_p} \cdot f_a \cdot Q_S$  **cientistas** e  $V_B \cdot \frac{PM_v}{PM_p} \cdot f_a \cdot Q_C$  **civis**. Já o vencedor perde  $V_B \cdot \frac{PM_p}{PM_v} \cdot Q_R$  **recursos** e  $V_B \cdot \frac{PM_v}{PM_p} \cdot f_a \cdot Q_M$  **militares**.

- $PM_a$  = poderio militar do atacante;
- $PM_d$  = poderio militar do defensor;
- $V_B$  = fator de violência da **batalha**;
- $PM_v$  = poderio militar do vencedor;
- $PM_p$  = poderio militar do perdedor;
- $f_d$  = fator de defesa (valor de balanceamento);
- $f_a$  = fator de agressividade da **espécie** do **povo**;
- $Q_S$  = quantidade de **cientistas**;
- $Q_C$  = quantidade de **civis**;
- $Q_M$  = quantidade de **militares**;
- $Q_R$  = quantidade de **recursos**;
- $Q_N$  = quantidade de naves;
- $Q_G$  = quantidade de combustível.

### 1.1.7 Exploração

A **exploração** é um mecanismo que permite a um **império** conhecer de forma imprecisa as características de um **planeta** sob o qual ele não tem controle. A **exploração** é feita por uma **colônia** sobre um **planeta** e, para um turno, devolve informações como o poderio militar, quantidade de população, quantidade de estruturas, fertilidade, atmosfera, água, temperatura e gravidade do **planeta**. Entretanto, como nunca é possível obter os dados com 100% de certeza, a **exploração** vem acompanhada de uma incerteza que define o quão precisos foram os dados obtidos.

Quando realizada **exploração**, num determinado turno, por uma determinada **colônia**, há incerteza no resultado, calculada através de características dessa **colônia**. Quanto mais **cientistas** possuídos pela **colônia**, junto com a quantidade de **instituições de ensino** (ambos fatores de auxílio nas descobertas de **tecnologia** e **explorações** da **colônia**), menor será a incerteza da exploração, retornando informações mais precisas em relação ao **planeta** explorado. Além disso, há um fator de aleatoriedade envolvido:

$$I = I_N - \left( \sum_{i=0}^n (Q_{IEi} \cdot E_{IEi}) + \sum_{j=0}^m \left( \frac{2 - f_{aj}}{100} \cdot Q_{Sj} \right) \right) \quad (1.3)$$

- $I$  = incerteza;
- $I_N$  = incerteza nativa;
- $n$  = quantidade de **instituições de ensino**;
- $Q_{IEi}$  = quantidade de **instituições de ensino**  $i$ ;
- $E_{IEi}$  = eficiência da **instituição de ensino**  $i$ ;
- $m$  = quantidade de **povos** cientistas;
- $f_{Aj}$  = fator de agressividade da **espécie** do **povo cientista**  $j$ ;
- $Q_{Sj}$  = quantidade de **cientistas** do **povo**  $j$ .

Exemplo: Nativamente, há uma incerteza de 70% nas explorações de uma colônia. Em marte, há 15 **instituições de ensino** “UFSCAR” com eficiência 0,4 e 10 **instituições de ensino** “USP” com eficiência 0,8. Também considera-se a existência de 1000 **cientistas** de agressividade 0,6 e 500 **cientistas** de agressividade 0,2 nessa **colônia**:

$$I = 70 - (15 \cdot 0,4) - (10 \cdot 0,8) - \left( \frac{2 - 0,6}{100} \cdot 1000 \right) - \left( \frac{2 - 0,2}{100} \cdot 500 \right) = 33\% \quad (1.4)$$

Finalmente, há uma checagem para evitar que  $I$  fique abaixo de 0%, sendo neste caso reajustado para o valor. Também há o fator de aleatoriedade na própria exploração, que é sorteado entre -10 e +10, adicionado na porcentagem total após o fim com as mesmas restrições de não negatividade anteriores. Dessa forma, no exemplo, a incerteza total das **explorações** feitas por marte poderia ficar em uma faixa entre 23% e 43%.



### 1.1.8 Recursos

São bens utilizados para realizar ações dentro do jogo. A exemplo, naves são movidas e combustível é queimado ao fazer **transporte**, melhor detalhado em seção própria. Já comida é descartável e mantém a população, armas são gastas em **batalhas** e matéria-prima é gasta para criar **estruturas**. São *possuídos* por um **planeta**, tornando-se **estoque**. O nível do recurso no estoque do planeta é dado pela tecnologia mais avançada conhecida pelos **cientistas** do planeta. Sua identidade é definida pelo nome, e são *precisos* para construção de **estruturas**. Para sua geração, pode *precisar* de qualquer número de **tecnologias**. São *gerados* por **atuação** dos **civis** nas **construções**. Seu volume é usado para calcular o custo de **transporte** desse recurso. O gasto de comida é melhor detalhado na seção de **povo**. Finalmente, é *priorizado* por vários **planetas** ao longo dos turnos, o que afeta a sua produção em cada tempo.

### 1.1.9 Prioridade

Existe para armazenar o que um determinado **planeta** tem mais necessidade de *gerar* naquele **turno**. A explicação do mecanismo de geração deste **recurso** é melhor documentada na seção sobre **construções**. Possui como atributo chave, junto ao seu turno, a ordem de prioridade, aquele **recurso** e seu **planeta**.

### 1.1.10 Estoque

O **estoque** é o local onde um **planeta** armazena certa quantidade de um **recurso** disponível para uso naquele determinado momento, turno. Cada **estoque** é composto por apenas um tipo de **recurso** e pode *ser transportado* entre **colônias** de um mesmo **império**. Para permitir esta mobilidade, são necessários alcance suficiente da **tecnologia** de **transporte** do **planeta** para viajar essa distância e naves e combustível suficientes.

### 1.1.11 Espécie

São compostas por múltiplas características, como volume, conformidade, trabalho, agressividade, faixa de gravidade, faixa de temperatura, faixa de água e faixa de atmosfera. As faixas simbolizam o alcance de valores (máximo e mínimo) nos quais esse povo pode sobreviver, enquanto trabalho e agressividade são utilizados, respectivamente, para cálculo de fatores relacionados à produtividade de **recursos** e militarização em **batalhas** de uma nação. Já a conformidade dita a tendência de um povo de sair do estado de **civil** para **militar** ou **cientista**. Além disso, **espécies** são identificadas pelo nome. O tamanho somente é usado para calcular o custo de transporte dessa espécie.

### 1.1.12 Povos

São identificados pela combinação de sua **espécie**, tipo, turno em que existem e o nome do **planeta** onde estão. Além disso, correm em determinada quantidade. Existem múltiplos **povos** diferentes em um mesmo **planeta**, todos juntos formando uma população. Todo **povo** *tem* somente uma **espécie**. **Povos** obrigatoriamente têm tipo, sendo, exclusivamente, ou **cientista**, ou **militar**, ou **civil**. Populações podem ser movidas de uma **colônia** para outra, mas, nesse momento, trocam de **povo**, assumindo a identidade da população do **planeta** de destino.

Em cada turno, a população cresce ou decresce baseada na quantidade de comida menos o número de população, diferença que nesta demonstração denomina-se saldo alimentício. Se esse valor for positivo em um turno, então a população cresce proporcionalmente à quantia. A parte da população que cresce sempre é **civil** (pois ninguém nasce **militar** ou **cientista**). A comida gasta sempre é igual à quantidade de população antes de aplicados crescimento e decrescimento, exceto quando o saldo alimentício é negativo, zerando-a.

Se o saldo alimentício for negativo, então a população decresce também proporcionalmente a este valor. Nesse caso, a porcentagem de **civis**, **militares** e **cientistas** de cada espécie que decresce é igual à proporção deste **povo** na população geral. Ou seja, em uma população onde há 20% de **cientistas**, 30% de **militares** e 50% de **civis**, caso a população decresça em 100 unidades, decrescerão 20 cientistas, 30 militares e 50 civis. A mesma distribuição ocorre em relação às espécies que compõem a população.

Abaixo equações que descrevem o comportamento citado acima:

$$P[t] = Q_S[t] + Q_M[t] + Q_C[t] \quad (1.5)$$

- $P$  = número da população;
- $Q_S$  = número de **cientistas**;
- $Q_M$  = número de **militares**;
- $Q_C$  = número de **civis**;
- $t$  = turno atual;
- $f_C$  = fator de crescimento geral;
- $f_D$  = fator de decrescimento geral.

O jogo possui um único fator de crescimento geral para todos os **povos**. O mesmo ocorre para o fator de decrescimento geral. Também:

Se  $F[t] \geq P[t]$ , ou seja, há mais comida do que população:

$$C_G[t] = f_C(F[t-1] - P[t-1]) \quad (1.6)$$

*Crescimento da população com disponibilidade de comida*

Se  $F[t] < P[t]$ , ou seja, há menos comida do que população:

$$P_d[t] = f_D(F[t-1] - P[t-1]) \quad (1.7)$$

*Decréscimo de população com falta de comida*

$$\frac{Q_C[t-1]}{P[t-1]} = \frac{Q_{Cd}[t]}{P_d[t]}, \quad \frac{Q_M[t-1]}{P[t-1]} = \frac{Q_{Md}[t]}{P_d[t]} \text{ e } \frac{Q_S[t-1]}{P[t-1]} = \frac{Q_{Sd}[t]}{P_d[t]} \quad (1.8)$$

*A proporção da população que decresce de cada tipo é igual à proporção deste tipo na população total*

- $F$  = quantidade de comida;

- $Q_{Cg}$  = quantidade de civis gerados;
- $P_d$  = quantidade de população decrescida;
- $Q_{Cd}$  = quantidade de **civis** decrescida;
- $Q_{Md}$  = quantidade de **militares** decrescida;
- $Q_{Sd}$  = quantidade de **cientistas** decrescida;

Mais detalhes sobre o crescimento e decréscimo de cada tipo da população é dado no texto referente àquele determinado tipo.

#### 1.1.13 Civil

Estes são uma especialização do **povo**. *Atuam* na **indústria** para *gerar recursos* e podem ser recrutados em uma **base militar** para gerarem **militares** ou estudar em uma **instituição de ensino** para gerarem **cientistas**. **Civis** de **espécies** menos agressivas geram mais recursos.

População de **civis** de um turno é dada por:

- **Civis** do turno anterior
- + **Civis** novos que reproduziram baseados no saldo alimentício
- - **Civis** que morreram de fome
- - **Civis** que morreram em **batalha** naquele turno

#### 1.1.14 Cientista

São uma especialização do **povo**. *Conhecem tecnologias* e são gerados pelo estudo dos **civis** na **instituição de ensino (Atuação)**. São utilizados para desenvolver **tecnologias**, múltiplas inovações para aumentar a produtividade e a quantidade de ações que um **império** pode realizar.

População de cientistas de um turno é:

- **Cientistas** do turno anterior
- + **Civis** que *atuaram* em **instituições de ensino** no turno anterior
- - **Cientistas** que morreram de fome
- - **Cientistas** que morreram em **batalha** naquele turno

#### 1.1.15 Militar

Outra especialização do **povo**. São gerados pelo recrutamento de **civis** na **base militar (Atuação)**. São utilizados no cálculo de poderio militar e morrem nas **batalhas**.

População de militares de um turno é:

- **Militares** do turno anterior
- + **Civis** que *atuaram* em **bases militares** no turno anterior

- - **Militares** que morreram de fome
- - **Militares** que morreram em uma **batalha** vencida naquele turno

Caso o **planeta** tenha perdido uma **batalha** naquele turno, sua população de **militares** é dizimada a zero.

#### 1.1.16 Estruturas

Possuem tipo, sendo, exclusivamente, ou **indústria**, ou **base militar**, ou **instituição de ensino**. Além disso, todas *precisam* de quantidades de **recursos** para serem construídas. No geral, cada **estrutura** possibilita diferentes ações dentro do jogo. Possuem lotação máxima, eficiência e, para reconhecê-las, nome.

#### 1.1.17 Instituição de Ensino

Nelas, *atua* uma quantidade de **civis** para se tornar **cientista**.

#### 1.1.18 Indústria

Nelas, *atuam* **civis** para gerarem **recursos**. Esta **atuação** tem uma quantidade de **civis** que a efetua. Ainda, toda **indústria** consegue gerar apenas um **recurso**.

#### 1.1.19 Base Militar

Nelas, *atua* uma quantidade de **civis** para se tornar **militar**.

#### 1.1.20 Atuação

A **atuação** é a interação de uma quantidade específica de **civis** com cada **construção**. A interação entre **civis** e uma **instituição de ensino** os tornam cientistas, a *interação* com a **base militar** faz virarem **militares** e a *interação* com a indústria gera **recursos** para a **colônia**. Tais interações são semânticas e, respectivamente, denominadas estudo, alistamento e trabalho.

Em todo turno as populações se movimentam para ganharem uma profissão. O fator de conformidade da **espécie** diz a porcentagem de um **povo civil** da população que tentará mudar e o fator de agressividade decide quantos % desses vão tentar ser **militares**, sendo que o resto tentará ser **cientista**. Entretanto, esse treinamento é limitado pela lotação máxima das **construções** de **instituições de ensino** e das **bases militares**, pois não se pode formar mais pessoas do que existem de vagas no **planeta**. A eficiência das **estruturas** não importa para essa geração.

Exemplo: **Planeta** com 2 universidades (60 vagas) e 3 bases militares (30 vagas). Supõe-se que há população de 1000 **civis** com 0,25 de conformidade e 0,6 de agressividade. Desses 1000, 250 vão querer parar de ser **civis**, sendo que 150 irão querer virar **militares** e 100 vão querer virar **cientistas**. Existem 120 vagas em universidade nesse **planeta**, então no final haverá 100 **cientistas**. Existem 90 vagas em **bases militares** nesse **planeta**, então no final terá 90 novos **militares** e 60 **civis** que não vão conseguir realizar seu sonho no exército.

A cada turno o jogador decide o foco industrial do momento, selecionando a quais **recursos** vai dar prioridade de geração. Seguindo isso, o jogo automaticamente aloca **civis** por meio de **atuação** para as **indústrias** que podem gerar esses **recursos**, dando prioridade para preencher

primeiro as indústrias com os **recursos** priorizados. Logo, dada uma **atuação** em uma **indústria**, a quantidade de **recursos** é dada pelo nível da **tecnologia** ligada ao **recurso**, a quantidade de **civis** e o fator de trabalho deles, a eficiência e quantidade da **indústria** e o fator de fertilidade do **planeta**.

$$Q = EP_m \cdot Nt! \cdot Q_C \cdot Ct \cdot E_I \cdot Q_I \cdot FP$$

- $EP_m$  = Eficiência produtiva base da **tecnologia** de **manufatura**;
- $Nt$  = Nível da **tecnologia**;
- $Q_C$  = Quantidade de civis envolvida (quantidade de **civis** nunca pode ser maior que lotação máxima da **estrutura** vezes a quantidade de **estruturas**);
- $Ct$  = Coef. de trabalho da **espécie**;
- $E_I$  = Eficiência da **construção**;
- $Q_I$  = Quantidade de **indústrias** envolvidas;
- $FP$  = Fertilidade do **planeta**.

### 1.1.21 Construção

Uma construção é uma **estrutura** que está em um **planeta** em certa quantidade. É possível, a qualquer momento, construir ou destruir **construções** de uma **colônia**. A sua destruição é instantânea e sem custos, mas não dá nada em troca, apenas espaço livre para uma nova. As construções existem em determinado turno para garantir histórico.

### 1.1.22 Tecnologia

É *conhecida* por uma população de **cientistas** (população de **cientistas** são todos os **povos cientistas** de um **planeta**, independente da **espécie**). Cada uma possui nome e nível, e sua combinação a identifica. Certa **tecnologia** precisa da mesma **tecnologia** de nível anterior para ser descoberta. Por exemplo, antes de desbloquear armamento 2, uma população de cientistas precisa ter descoberto armamento 1. **Recursos** *precisam* de *tecnologias* para serem gerados. Seguindo o exemplo anterior, é preciso *conhecer* armamento 1 antes de ser possível *gerar* **recursos** armamentistas. **Tecnologias** devem ter um de 3 nomes: **transporte**, **armamento** e **manufatura**. É importante ressaltar que o nível 0 de todas é sempre conhecido e, após conhecer um novo nível de certa **tecnologia**, o anterior torna-se ultrapassado e não é mais utilizado.

A complexidade de certa **tecnologia** vai de 0 a 1 e é um fator que dita a chance de ser descoberta, sendo que, quanto mais alto, mais simples é a **tecnologia**.

Todo turno um **planeta** tem uma chance de descobrir uma **tecnologia**. Somente o próximo nível das **tecnologias** já conhecidas pode ser descoberto, com chance de ser descoberta baseada na complexidade dela, na quantidade de **cientistas** do **planeta** e suas agressividades, na quantidade de **instituições de ensino** e suas eficiências.

A ciência em um planeta é comunicativa, pois quando um **cientista** *descobre* uma **tecnologia** todos daquele **planeta** (não importando a **espécie**) também a aprendem.

$$C_t = \left( \frac{\sum_{i=0}^n (1 - f_{Ai}) \cdot Q_{Si}}{\sum_{i=0}^n Q_{Si}} + \frac{\sum_{i=0}^k E_{IEi} \cdot Q_{IEi}}{\sum_{i=0}^k Q_{IEi}} \right) \cdot f_{ct}$$

Onde:

- $C_t$  = probabilidade de descobrir a **tecnologia**;
- $n$  = quantidade de **povos**;
- $f_{Ai}$  = fator de agressividade do **povo**  $i$ ;
- $Q_{Si}$  = quantidade de **cientistas** do **povo**  $i$ ;
- $k$  = quantidade de **estruturas** do tipo **instituição de ensino**;
- $E_{IEi}$  = eficiência da estrutura **instituição de ensino**  $i$ ;
- $Q_{IEi}$  = quantidade de **construções** da **estrutura**  $i$ ;
- $f_{ct}$  = fator de complexidade da **tecnologia**.

Exemplo numérico: Planeta com dois povos. Povo A: 0,56 de agressividade e 300 cientistas. Povo B: 0,32 de agressividade e 800 cientistas. **Tecnologias** conhecidas pela comunidade acadêmica: **armamento 1**, **manufatura 2** e **transporte 1**. Dessa maneira, as próximas **tecnologias** que podem ser descobertas são: **armamento 2** (0,7 de complexidade), **manufatura 3** (0,43 de complexidade) e **transporte 2** (0,6 de complexidade). Existem 4 **instituições de ensino** "uniesquinas" com 0,35 de eficiência, 2 "institutos federais" com 0,45 de eficiência, 2 "UFSCAR" com 0,55 de eficiência e 1 "USP" com 0,7 eficiência. Portanto:

Chance de descobrir **armamento 2**:

$$\left( \frac{(1 - 0,56)300 + (1 - 0,32)800}{1100} + \frac{4 \cdot 0,35 + 2 \cdot 0,45 + 2 \cdot 0,55 + 1 \cdot 0,7}{9} \right) \cdot 0,7 = 0,1959 = 19,59\%$$

Chance de descobrir **manufatura**:

$$\left( \frac{(1 - 0,56)300 + (1 - 0,32)800}{1100} + \frac{4 \cdot 0,35 + 2 \cdot 0,45 + 2 \cdot 0,55 + 1 \cdot 0,7}{9} \right) \cdot 0,43 = 0,1203 = 12,38\%$$

Chance de descobrir **transporte 2**:

$$\left( \frac{(1 - 0,56)300 + (1 - 0,32)800}{1100} + \frac{4 \cdot 0,35 + 2 \cdot 0,45 + 2 \cdot 0,55 + 1 \cdot 0,7}{9} \right) \cdot 0,6 = 0,1679 = 16,79\%$$

### 1.1.23 Manufatura

Uma **manufatura** tem sua eficiência produtiva, influenciando em quanto de cada recursos um **planeta** consegue produzir.

#### 1.1.24 Armamento

Uma **tecnologia de armamento** tem seu poder de destruição, que influencia no poderio militar daquela colônia ao afetar o nível do armamento no estoque.

#### 1.1.25 Transporte

Uma **tecnologia de transporte** tem sua capacidade de transporte e alcance, os quais definem, respectivamente, quanta carga e o quão longe vão às naves do **planeta**. Este mecanismo é melhor exemplificado na próxima seção.

#### 1.1.26 Movimento de estoque e povo

Um **império** pode efetuar transferência de **estoque** e **povo** entre duas de suas **colônias**. A quantidade de um **recurso** ou população movimentados é limitada pela disponibilidade de naves e combustível no **estoque** da **colônia** de origem. Outro fator que limita esta quantia é a capacidade de transporte das naves, definida pela **tecnologia de transporte**, dado que tanto **espécie** quanto **recurso** possuem tamanho. O **recurso** de naves usado para o transporte fica no **estoque** do **planeta** de destino e é removido do **estoque** de origem após a viagem (sendo o seu nível tecnológico atualizado para o do **planeta** destino). Todavia, o combustível utilizado é descartado. No caso de *transporte* de **povo**, é necessário checar a capacidade populacional do **planeta** de destino, além de se a **espécie** sobrevive às suas condições. Outra consideração a se fazer no *transporte* de **povo** é que, caso seu tipo seja **cientista**, o conhecimento de **tecnologias** é transportado junto e transmitido para os **cientistas** do **planeta** destino, atualizando todo o seu **estoque**.

### 1.2. Estratégia Geral

Para desenvolver uma estratégia geral de jogo, é importante considerar os seguintes elementos e objetivos:

#### 1.2.1 Fase de Preparativos (Primeira Fase)

- Execute ações de gerenciamento **colonial**, como recrutamento, estudo, trabalho e construção, em cada colônia.
- Priorize a construção de **indústrias** para aumentar a geração de **recursos** e **instituições de ensino** para formar **cientistas**.
- Desenvolva **tecnologias**, especialmente aquelas relacionadas à **manufatura** e **armamento**, para aumentar a produtividade e o poderio militar.

#### 1.2.2 Fase de Cooperação Imperial (Segunda Fase)

- Realize ações de cooperação imperial, como movimentar recursos e população entre **colônias** e explorar novos **planetas**.
- Concentre-se em aumentar a eficiência das suas **colônias** e na expansão para novos **planetas**. **Exploração** é fundamental para encontrar **planetas** adequados para colonização.

- Mantenha um equilíbrio entre **recursos**, população e poderio militar em cada **planeta** para garantir a segurança do seu **império**.

### 1.2.3 Fase de Batalha (Terceira Fase)

- Prepare-se para **batalhas**. Realize ataques estratégicos em outros **planetas** para expandir seu território.
- Avalie a força militar do inimigo e escolha alvos que possam ser conquistados com sucesso. Lembre-se de que a superioridade militar é essencial para a colonização.
- Gerencie seu poderio militar com eficiência, recrutando mais **militares** quando necessário e mantendo um **estoque** de armas adequado.

### 1.2.4 Colonização e Domínio:

- *Colonize* **planetas** conquistados para expandir seu **império**. Cada nova **colônia** contribui para seus objetivos.
- Lembre-se de que o jogo continua até que um **império** tenha colonizado 100% dos **planetas** disponíveis, então mantenha o ritmo de expansão constante.
- Esteja preparado para defender suas **colônias** conquistadas, pois outros jogadores podem tentar retomá-las.

### 1.2.5 Desenvolvimento Tecnológico

- Priorize o desenvolvimento tecnológico para aumentar sua eficiência e capacidade de ação.
- Adquira **tecnologias** de **transporte**, **manufatura** e **armamento** para melhorar suas capacidades logísticas e militares.
- Utilize **cientistas** para acelerar a pesquisa de novas **tecnologias** e inovações.

### 1.2.6 Diplomacia e Conflito

- Esteja aberto à diplomacia e à cooperação com outros jogadores quando for do seu interesse.
- Esteja preparado para confrontos e conflitos quando necessário para alcançar seus objetivos de expansão.

### 1.2.7 Gerenciamento de Recursos

- Monitore constantemente seus **estoques** de **recursos** e equilibre o uso de **recursos** para manter sua economia em funcionamento.
- Certifique-se de que suas **colônias** tenham acesso a **recursos** suficientes para manter a produção e a população.

Lembrando que a incerteza na exploração de **planetas** e o desenvolvimento de estratégias adaptativas são essenciais para ter sucesso no jogo "Intergalactic Empires". A estratégia geral deve ser flexível e ajustada conforme a dinâmica do jogo e as ações dos outros jogadores.



### 1.3. Notas do Modelo

#### Identificadores "secundários"

- 1 - Coordenada também pode ser usada como chave de **planeta**.
- 2 - Cor é **única** chave secundária para **império**.

#### Chaves de agregações

- 3 - Chave de **Colônia** é formada por **império**, **planeta** e turno de início.
- 4 - Chave de **Batalha** é formado por **colônia** atacante, **planeta** atacado e o turno da **batalha**.
- 5 - Chave de **estoque** é formada por **planeta**, **recurso** e turno.
- 6 - Chave de **estoque** é formada por **planeta**, **recurso** e turno.
- 7 - Chave de **construção** é formada por **estrutura**, **planeta** e turno.
- 8 - Chave de **atuação** é formada por **construção** e **civil**.
- 9 - Chave de **exploração** é formada por **império**, **planeta** e turno.
- 10 - Chave de prioridade é formada por **planeta**, **recurso**, ordem e turno. Não pode ter um determinado recurso, num determinado planeta, em um determinado turno com a mesma ordem.

#### Restrições de integridade e Funcionalidades que não podem ser garantidas no MER

- 11 - **Estoques** de **recursos** só podem ser movidos pelo **império** que governa o **planeta** daquele **estoque** para uma **colônia** diferente daquele **império**.
- 12 - Caso a **estrutura** seja **base militar**, todos os **civis** viram **militares**, nesse caso remover **civis** e adicionar **militares**. Caso seja **Inst. Ensino**, viram **cientistas**, nesse caso remover **civis** e adicionar **cientistas**.
- 13 - **Planeta** e turno de **civil** e **construção** em uma **atuação** precisam ser iguais. O que é possível de ser feito usando um check em um SGBD.
- 14 - Após gerar **recurso** é preciso atualizar o **estoque** do **planeta** onde ocorreu a atuação. Esse relacionamento só é valido quando a **estrutura** da **construção** da **atuação** é válida.
- 15 - Ciclo **Colônia - Planeta**: Não é possível uma **colônia** *explorar* a si mesma ou *explorar* **planetas** que sejam **colônias** do próprio **império**.
- 16 - Uma **tecnologia** só depende da **tecnologia** de mesmo nome e nível anterior.
- 17 - Dado o nome e um nível de uma **tecnologia**, um **cientista** só pode *conhecê-la* se ele *conhece* a **tecnologia** de mesmo nome e nível anterior.
- 18 - Ciclo **Colônia - Planeta**: Uma **colônia** não pode *batalhar* com um **planeta** que é **colônia** do mesmo **império**.

- **19** - Quantidade de **civis** em uma **atuação** não pode ser maior que quantidade de **construções** em que aquela **atuação** acontece vezes a lotação máxima da **estrutura**.
- **20** - É necessário ter naves o suficiente no **planeta** da população para levar a quantidade de população desejada e, além disso, a nave tem que ser tirada do **estoque de recursos** do **planeta** de onde saiu. Essa informação vem de "Qtd nave"
- **21** - Para o **recurso** ser gerado o **planeta** precisa ter acesso à **tecnologia** que permite gerar ele.
- **22** - Um **povo** só pode ser movido pelo **império** que governa aquele **povo** para uma **colônia** diferente daquele mesmo **império**.
- **23** - O **povo** que *morre* precisa ser de um dos **planetas** envolvidos na **batalha**.
- **24** - É necessário garantir a consistência do turno de **batalha** com o turno de **povo**.
- **25** - É necessário garantir a consistência do *turno* de **batalha** com o turno de **estoque**.
- **26** - O **recurso** gasto precisa ser de um dos **planetas** envolvidos.
- **27** - Não pode morrer mais população do que um **povo** tem.
- **28** - Não pode consumir mais **recursos** do que um **planeta** tem.
- **29** - O *vencedor* sempre precisa ser participante de uma **batalha**.
- **30** - O poderio militar do defensor é igual ao poderio militar do **Planeta** e o poderio militar do atacante é igual ao poderio militar da **Colônia**
- **31** - Quando **cientistas** são movidos eles levam todas as **tecnologias** que sabem. Nesse caso, deve-se atualizar a tabela de **tecnologias**.
- **32** - Um **cientista** aprende todas as **tecnologias** conhecidas por populações vivas de **cientistas** do **planeta** de origem. Nesse caso, deve-se atualizar a tabela de **tecnologias**.
- **33** - Os **povos** que chegam em **planetas** onde sua **espécie** foi dizimada tem acesso ao conhecimento perdido de sua **espécie**.
- **34** - Uma **estrutura** do tipo **industria** não pode precisar do tipo de **recurso** que ela gera para ser construída.
- **35** - Dado uma **atuação**, só é possível gerar o **recurso** que a **indústria** envolvida consegue gerar.
- **36** - O relacionamento de "gera recurso" precisa estar ligado a uma **atuação** em **construções** onde a **estrutura** é do tipo **indústria**, sendo necessário garantir essa consistência em aplicação.
- **37** - Quando um **povo** descobre uma **tecnologia**, os **cientistas** de todas as **espécies** daquele **planeta** também descobrem a **tecnologia**.

## 1.4. Principais operações

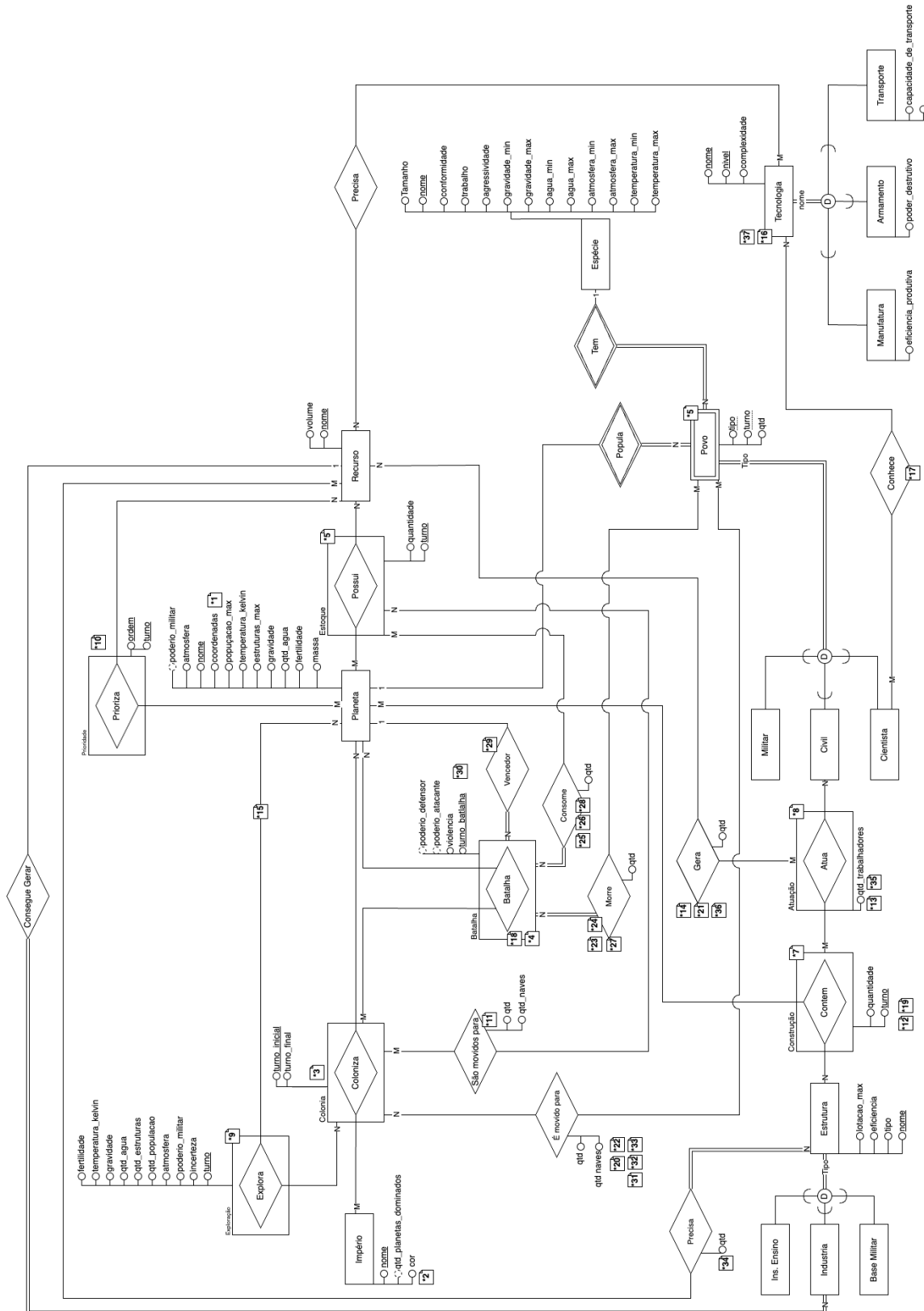
### 1.4.1 Império

- **Explorar um planeta:** inserir um **planeta** na lista de **planetas** explorados.
- **Colonizar um planeta:** inserir um **planeta** na lista de **planetas colonizados**.
- **Batalhar com um planeta:** cria uma **batalha** na tabela e atualiza os gastos de **recursos** e **povo**.
- **Construir e destruir estruturas:** atualizar o número de **estruturas** de um **planeta**.
- **Transportar povo e estoque:** atualizar o número de **povo** e **estoque** das duas **colônias** envolvidas, levando em conta os **recursos** utilizados no procedimento.
- **Priorização de recursos:** Inserir novas **prioridades** do turno ou duplicar as **prioridades** do *turno* anterior.

### 1.4.2 Consultas

- População total de um **império** podendo filtrar por tipo de **povo** e **espécie**
- População total de um **planeta** podendo filtrar por tipo de **povo** e **espécie**
- **Tecnologias** conhecidas pelo **império**
- Quantidade de **recursos** no **estoque** de **planetas** e **impérios**
- **Espécies** podendo filtrar por **planeta** ou **império**
- Histórico de **povo** podendo filtrar por turno, **planeta** e **império**
- Histórico de *explorações* podendo filtrar por **planeta** ou **império**
- Histórico de **construções** podendo filtrar por **planeta** ou **império**
- Histórico de dominação de um **planeta** podendo filtrar por turno
- Histórico de dominações de um **império** podendo filtrar por turno
- Priorização de **recursos** de um **planeta**
- Histórico de **atuações** podendo filtrar por **planeta** ou **império**

## 2. MODELO ENTIDADE-RELACIONAMENTO



### 3. MUDANÇAS REALIZADAS EM RELAÇÃO À PRIMEIRA ENTREGA

#### 3.1. Na descrição do Problema e dos Requisitos de Dados

O texto foi alterado para deixar mais claros os requisitos do sistema. As menções ao histórico de turnos foram removidas de certas seções (e adicionadas em outras), juntamente com informações e descrições de mecanismos removidos do MER-X. A estratégia geral também foi melhor descrita e segmentada em fases. Descrições sobre atributos e mecanismos que antes faltavam ou davam vez para ambiguidades ou superficialidades foram melhorados, simplificando certas coisas e detalhando outras muito mais em prol de maior especificidade. As notas do modelo também foram adicionadas ao texto e removidas da figura do MER-X devido a seu tamanho exacerbado. No geral, o detalhamento e comprimento do texto quadruplicou para especificar os mínimos detalhes e corrigir minuciosamente observações feitas pela correção.

#### 3.2. No MER-X

Certas nomenclaturas e erros de digitação mudaram. Como uma expedição de **exploração** é feita a partir de um **planeta** específico do **império**, o relacionamento "Explora" agora parte de uma **colônia**, e não do **império**. A cardinalidade do relacionamento coloniza foi alterada de 1:N para N:M, já que vários impérios podem colonizar um mesmo planeta ao longo do jogo. Pela mesma lógica, a cardinalidade de "batalha" foi alterada de 1:1 para N:M. Além disso, múltiplos atributos com detalhes sobre um **planeta** explorado agora fazem parte da agregação **exploração**.

O relacionamento "move população" deixa de ser ternário envolvendo duas **colônias** para ser binário envolvendo apenas uma, a **colônia** de destino. Isso dado que é possível inferir a **colônia de origem** já que **povo** agora é entidade fraca de **planeta**. Também foi removida a agregação deste relacionamento, pois, como **povo** já possui turno, o chaveamento de agregação com turno se torna redundante e uma fonte de inconsistência. Pelos mesmos motivos, essas alterações também foram feitas ao relacionamento "move recurso". Tais modificações criaram a possibilidade de inconsistências, marcadas em comentários. Porém, percebeu-se a importância dos históricos de turnos em **estoque** e **construção**, já que demonstram a evolução do jogador ao longo do jogo. Por isso, nesses casos foi adicionado atributo turno como parte da chave das agregações.

Para armazenar o **planeta** vencedor de uma **batalha**, o relacionamento "vencedor" foi criado. A cardinalidade do relacionamento "precisa" entre **recurso** e **estrutura** foi modificada de 1:N para N:M, já que essa pode precisar de mais de um tipo daquele. A participação para **estrutura** agora é obrigatória, pois toda precisa de **recursos**. Agora, o **planeta** *prioriza* um **recurso** para *gerar*, formando um novo relacionamento e agregação.

**Tecnologia de Transporte** possui agora seu atributo alcance, que depende do nível em que está. Uma das maiores alterações foi a simplificação em como **tecnologias** são evoluídas. Essencialmente, existem somente 3 tipos de **tecnologia**, com múltiplos níveis cada. Uma **tecnologia** do nível A só precisa da mesma **tecnologia** de nível A - 1 para existir, seguindo a lógica de aprimoramentos (nível 5 precisa do 4, que precisa do 3...). Assim, pela simplicidade desse tratamento em aplicação, não há necessidade do auto-relacionamento binário "precisa" em **tecnologia**, que poderia causar inconsistências, sendo removido.

**Império** ganhou o atributo cor, que também o identifica. Agora, toda **batalha** *mata povos* e *consome estoques*. **Batalha** também ganhou o atributo de violência da batalha, além dos derivados poderio militar do atacante e poderio militar do defensor. **Espécie** e **Recurso** agora têm o atributo tamanho de forma a modelar o transporte deles por naves.

Em **espécie**, todo atributo do tipo "faixa de valores" foi aberto em dois novos atributos, os valores máximo e mínimo desta faixa. Alterou-se **povo** para ser também entidade fraca de **espécie** devido à semântica do problema, já que não pode existir sem ela. Logo, a participação do relacionamento entre **povo** e **espécie** mudou para total do lado daquele. A maior modificação foi que todo o esquema de operação de profissões e **estruturas** de formação delas foi simplificado, com nova documentação massiva. As agregações **estudo**, **trabalho** e **recrutamento** foram removidas e substituídas pela agregação **atuação**, que, diferente das anteriores, não possui turno (era redundante e podia causar inconsistência). Logo, os relacionamentos "gera" em **militar** e **cientista** foram removidos. Agora, **instituição de ensino** e **base militar** não possuem relacionamentos específicos, e **indústria** pode gerar **recursos**. O **Civil** atualizado se relaciona com a **construção**, e não com **indústria**. Finalmente, as **estruturas** agora têm geração instantânea, sendo removido o tempo de **construção** por dificuldades no MR. As modificações no sistema de geração de profissões têm objetivo de simplificar e aumentar a eficiência geral do programa, mantendo a lógica principal de formação das profissões.



## 4.2. Justificativas dos Mapeamentos

### 4.2.1 Império

Quantidade de **planetas** dominados foi modelado, pois é uma informação simples de ser calculada e comumente consultada.

Como cor também é identificador e todo **império** tem, uma chave secundária que nunca é nula foi criada para garantir unicidade.

### 4.2.2 Planeta

Poderio militar é um cálculo complexo, logo faz sentido tê-lo salvo o tempo todo para economizar consultas. Entretanto, é importante garantir que todo turno o poderio de todos os **planetas** seja devidamente atualizado.

Como coordenada também é identificador e todo **planeta** possui, há chave secundária não nula para garantir essa unicidade.

### 4.2.3 Prioridade de produção

O chaveamento da agregação, incluindo **planeta**, **recurso**, ordem e turno, permite que a prioridade de produção mude a cada turno. Para garantir que não ocorra repetição de ordem de prioridade num mesmo turno, há uma chave secundária.

### 4.2.4 Povo

Considerando que a entidade generalizada e somente uma entidade especializada possui relacionamentos relevantes (**Civil**), apenas eles foram modelados, se diferenciando dos mapeamentos padrões. Uma opção alternativa seguindo o mapeamento padrão seria modelar todas as entidades especializadas, mas julgou-se que a criação das tabelas de **cientistas** e **militares** não teriam utilidades práticas para justificarem suas existências.

A maioria dos relacionamentos está no **povo**, o qual, como precisa ter chaveamento no tipo, é possível salvar as quantidades diretamente. Logo, faz sentido mapear a tabela de **povo**.

Como **militar** não possui relacionamentos nem atributos, não faz sentido a criação de tabela própria.

Apesar de **cientista** possuir relacionamento, a semântica geral do projeto sugere que o **povo** *conhece* a **tecnologia**, sendo a unicidade do **cientista** o seu movimento, em que a **tecnologia** é levada para o **planeta** destino. No momento em que houver a movimentação de **cientistas** entre **colônias**, é necessário atualizar a tabela de **tecnologias**.

A chave de **civil** não inclui tipo, pois, como é uma disjunção, todas as tuplas que estarão na tabela de **civil** têm atributo *Tipo = "Civil"*, não faz sentido ter tipo na chave. Assim, podemos excluir *Tipo* da chave de civil, tornando a consulta mais rápida ao diminuir o tamanho do índice. Logo, a existência desse atributo nem seria necessária para o jogo, mas é obrigatório referenciar a chave estrangeira inteira. Esse mapeamento garante participação total para civil.

### 4.2.5 Conhecimento

Na tabela conhecimento, é necessário garantir que o tipo sempre é **cientista**. Como sempre *Tipo = Cientista*, o atributo não é parte da chave. Também não faz sentido armazenar uma



**espécie** do mesmo **planeta** conhecendo a mesma **tecnologia** em turnos diferentes, turno também é removido da chave.

Esse modelo não garante que um **cientista** só conhece uma **tecnologia** caso conheça a **tecnologia** de mesmo nome e nível anterior, sendo necessário tratar isso em aplicação.

A ciência em um planeta é comunicativa, pois quando um **cientista** *descobre* uma **tecnologia** todos daquele **planeta**, não importando a **espécie**, também a aprendem. Portanto, é preciso atualizar a tabela de conhece tecnologia com a **tecnologia** para todos os **povos**.

#### 4.2.6 Estrutura

Como não há atributos nas especializações e apenas um relacionamento, optou-se por economizar tabelas e mapear o CEG e somente o CEE de **indústria**, que precisa gerar **recurso**. A opção alternativa seria mapear todas as entidades especializadas, como sugere o mapeamento padrão, mas as tabelas de **Instituições de ensino** e **bases militares** não teriam utilidades práticas.

Como estruturas do tipo **indústria** sempre *geram* apenas um **recurso**, **recurso** gerado vira um atributo em sua tabela, com valor não nulo e fora da chave primária da tabela. Dessa forma, a cardinalidade é garantida e toda **estrutura** do tipo **indústria** presente na tabela **indústria** tem a participação total garantida. Caso fosse utilizada a opção de uma nova tabela, essa participação total não poderia ser garantida.

No mapeamento de **estrutura** *precisa* de **recursos**, não é possível garantir a participação total da **estrutura** no modelo relacional

#### 4.2.7 Tecnologia

Para mapear **tecnologia** escolheu-se criar apenas uma tabela. Isso ocorre ao passo que, como existirá uma quantidade fixa e pequena de tuplas, o sistema vai economizar o overhead de tabelas que não teriam utilidade prática. Entretanto, isso traz uma quantidade de atributos nulos que precisam ter a consistência garantida por aplicação, já que os atributos de tipo precisam concordar com o atributo de critério nome.

A outra opção seria fazer uma tabela para cada especialização e outra para a generalização, mas, como dito anteriormente, haveria alto overhead de mais três tabelas. Isso só seria interessante caso o existissem muitos níveis para cada tecnologia.

Ainda é preciso tratar em aplicação o requisito de que uma **tecnologia** com um certo nome e nível precisa da **tecnologia** de seu equivalente de nível anterior para ser acessada.

#### 4.2.8 Estoque

Como é desejado salvar o histórico de evolução do **estoque** de um determinado **planeta**, adiciona-se turno como chave de **estoque** para salvar a quantidade de um determinado **recurso** presente no **planeta** em um determinado turno.

#### 4.2.9 Construção

Em **construção** também deseja-se salvar seu histórico de evolução em um determinado **planeta**. Logo, também adiciona-se turno como chave, sendo possível acessar, para um determinado turno, num determinado **planeta**, a quantidade de **estruturas** presentes nele.

#### 4.2.10 Colônia

Como um **planeta** pode mudar de **império** dominante, é necessário que se saiba o turno de início da dominação. Além disso, é importante garantir que, quando uma troca acontece, o atributo turno final é devidamente preenchido para armazenar consistentemente o período de dominação do **império** que estava controlando o **planeta**.

#### 4.2.11 Exploração

Deseja-se salvar o histórico de **explorações** realizadas por uma determinada **colônia**, adicionando turno como chave, sendo possível acessar, para uma determinada **colônia** e **planeta**, o turno em que a **exploração** aconteceu. Isso permite que uma **colônia** *explore* um **planeta** várias vezes.

É necessário garantir que a **colônia** não explore **planetas** do próprio **império**.

#### 4.2.12 Batalha

É necessário que turno componha a chave, pois é possível que uma **colônia** batalhe o mesmo **planeta** diversas vezes (uma vez por turno).

Uma **batalha** não pode acontecer entre **planetas** que, naquele turno, estejam sob a dominação de um mesmo **império**.

É importante garantir que o **planeta** vencedor seja um dos envolvidos na **batalha**, e, como sempre há um planeta vencedor, o atributo tem que ser não nulo, garantindo a participação total.

O planeta\_atacante é sempre proveniente de uma **colônia**, e o planeta\_defensor sempre de **planeta**, e, conseqüentemente, o mesmo acontece com poderio\_militar\_atacante e poderio\_militar\_defensor. Optou-se por incluir o poderio militar dos planetas envolvidos, pois todas às vezes que se falar sobre uma batalha se quer saber qual o poderio militar dos planetas envolvidos naquela batalha, assim visamos economizar nas consultas para saber apenas dois pontos de dados.

Optou-se pela adição de um ID sintético em **batalha**, pois a chave primária anteriormente formada estava grande, assim acelerando consulta e diminuindo o index.

#### 4.2.13 Estoque\_gasto\_batalha

O **planeta** do **estoque** gasto tem que ser necessariamente um dos **planetas** envolvidos na **batalha**.

O turno do **estoque** precisa estar consistente com o turno da **batalha**, uma vez que não faz sentido gastar um **estoque** de um momento diferente do qual está acontecendo a **batalha**.

Não é possível gastar mais **recursos** que o **estoque** possui.

Uma **batalha** sempre gasta **estoque**, porém essa participação total não é possível de ser garantida no modelo relacional.

#### 4.2.14 povo\_morto\_batalha

O **planeta** que teve o **povo** morto tem que ser necessariamente um dos **planetas** envolvidos na **batalha**.

O turno do **povo** precisa estar consistente com o turno da **batalha**.

Não é possível morrer uma quantidade maior de **povo** que o **planeta** possui.

Em uma **batalha** sempre há mortos, porém, essa participação total não pode ser garantida no modelo relacional.

#### 4.2.15 Atuação

**Planeta** do **civil** e **planeta** da **construção** precisam ser iguais, do mesmo modo o turno de **civil** precisa ser o mesmo turno de **construção**. Nos dois casos é possível realizar um check em SGBD para garantir a consistência. Como a chave de **atuação** acaba sendo grande, poderia ter sido colocado ID em **construção** e **civil**, mas isso seria ineficaz, pois como é necessário garantir a igualdade de **planeta** e turno, possuí-los em uma única tabela só torna bem mais rápido e prático.

Como **planeta** e turno de **civil** sempre é igual ao **planeta** e turno da **construção** e temos um check que garante isso, optamos por remover **planeta** e turno da **construção** da chave. Isso diminui o tamanho da chave, acelerando as consultas, sem perder a unicidade das tuplas. Esse mapeamento não é usual, mas, após muita reflexão, julgou-se ser a melhor opção.

Para **atuações** onde a **estrutura** é **base militar**, todos os **civis** envolvidos viram **militares** e é preciso atualizar isso na tabela de **povo**.

Para atuações onde a **estrutura** é **instituição de ensino**, todos os **civis** envolvidos viram **cientistas** e é preciso atualizar isso na tabela de **povo**.

É preciso garantir que a quantidade de **civis** não seja maior que a lotação máxima da **construção** (lotação máxima da **estrutura** multiplicada pela quantidade de **estruturas**).

#### 4.2.16 Gera\_recurso

É preciso atualizar a tabela de **estoque** do **planeta** da **atuação** com os novos **recursos**.

Para o **recurso** ser gerado o **planeta** precisa ter acesso à **tecnologia** que o permite *gerar*.

Dada uma **atuação**, só é possível *gerar* o **recurso** que a **indústria** envolvida consegue *gerar*.

O relacionamento de gera\_recurso precisa estar ligado a uma **atuação** em **construções** onde a **estrutura** é do tipo **indústria**, sendo necessário garantir essa consistência em aplicação.

#### 4.2.17 move\_recurso

**Estoques** de **recursos** só podem ser movidos pelo **Império** que governa o **planeta**, daquele **estoque** para uma **colônia** diferente do mesmo **império**.

É necessário garantir que a **colônia** tenha naves o suficiente no **planeta** do **estoque** para levar a quantidade de **recurso** desejado e, além disso, a nave tem que ser removida do **estoque** do **planeta** de onde saiu e atualizada no **estoque** do **planeta** onde chega. Assim é preciso garantir a consistência do atributo qtd\_nave.

#### 4.2.18 move\_povo

Um **cientista** aprende todas as **tecnologias** conhecidas por populações vivas de **cientistas** do planeta de **destino**, sendo necessário atualizar a tabela de **tecnologias**.

Um **cientista** leva todas as **tecnologias** conhecidas por ele para o **planeta** novo.

Os **povos** que chegam em **planetas** onde sua **espécie** foi dizimada tem acesso ao conhecimento perdido de sua **espécie**, e isso ocorre naturalmente no modelo.

Um **povo** só pode ser movido pelo **império** que governa aquele **povo** para uma **colônia** diferente daquele mesmo **império**, portanto, é preciso garantir que **planeta** da **colônia** e **planeta** do **povo** sejam ao diferentes.

É necessário garantir que a **colônia** tenha naves o suficiente no **planeta** do **povo** para levar a quantidade de seres desejada e, além disso, a nave tem que ser removida do estoque do planeta de onde saiu e atualizada no **estoque** do **planeta** onde chega. Assim, é preciso garantir a consistência do atributo `qtd_nave`.

## 5. MUDANÇAS REALIZADAS EM RELAÇÃO À SEGUNDA ENTREGA

### 5.1. No relatório

Erros de digitação e formatação foram corrigidos, além da nomenclatura de muitos atributos para adequá-la ao padrão utilizado no MR e código SQL.

### 5.2. No DER

O atributo massa foi adicionado ao Planeta, . A notação de turno na entidade Povo também foi corrigida para ser chave. No geral, múltiplos atributos foram ajustados para terem o mesmo nome dentre DER, MR e código SQL.

### 5.3. No MR

O atributo massa do Planeta também foi adicionado no MR, além de um ID sintético na entidade. Atuação Houve modificação dos nomes dos atributos Planeta\_Atacado e turno para planeta\_defensor e turno\_batalha na tabela Batalha. Já na tabela Exploração, o atributo recurso foi modificado para fertilidade para manter acordo com o DER, além da adição do atributo violência. O atributo turno também foi acrescentado à tabela move\_recurso. Na tabela Povo, a notação de notnull foi acrescentada no atributo qtd. Muitos atributos tiveram nomes levemente modificados para entrar em acordo com o código SQL da terceira parte. Também foram melhoradas as justificativas sobre a chave da entidade Civil e o mapeamento dos atributos derivados em Batalha. Além disso, foram removidos quaisquer tipos de acentuação nos nomes de atributos/tabelas para a sintaxe ficar idêntica ao código SQL.

## 6. CONSULTAS DO PROJETO

No contexto do projeto, oito consultas foram estabelecidas:

### 6.1. Quantidade de Recursos de dado Império em um dado Turno

```

1 SELECT
2     R.NOME,
3     NVL(SUM(E.QUANTIDADE), 0) AS QUANTIDADE
4 FROM
5     ESTOQUE E
6     JOIN COLONIA C
7         ON E.TURNO = :TURNO.ATUAL
8         AND E.PLANETA = C.PLANETA
9         AND C.IMPERIO = :IMPERIO AND C.TURNO_INICIAL <= :TURNO.ATUAL
10        AND (C.TURNO_FINAL >= :TURNO.ATUAL OR C.TURNO_FINAL IS NULL)
11     RIGHT JOIN RECURSOS R
12         ON E.RECURSO = R.NOME
13 GROUP BY
14     R.NOME
15 ORDER BY
16     R.NOME
17 ;

```

Localização: */database/consultas.sql*

Nesta consulta, a quantidade de cada recurso possuído por um império é impresso em cada linha. Para isso, as tabelas estoque, colônia e recurso são juntadas para mesmo planeta e recurso. Utilizando a variável império alimentada à consulta, bem como o turno atual, filtram-se apenas colônias pertencentes ao império no determinado turno. Finalmente, há um agrupamento por recurso para impressão correta e utilização da soma de todos os recursos daquele tipo em cada planeta/colônia. A ordenação também é feita por recurso.

Tal consulta é extremamente importante no contexto do jogo, pois é através dos recursos que os jogadores efetuam ações dentro do jogo. É, logo, importante conseguir buscar esta informação a qualquer instante para que o império possa ter maior noção e controle do plano geral de suas posses ao, por exemplo, escolher prioridade de produção e movimentação de recursos.

### 6.2. Quantidade de Recursos Produzidos por um dado Império até Dado Turno

```

1 SELECT
2     R.NOME AS RECURSO,
3     NVL(SUM(G.QTD), 0) AS QUANTIDADE
4 FROM
5     GERA_RECURSO G
6     JOIN ATUACAO A
7         ON A.TURNO_CIVIL <= :TURNO
8         AND G.ID_ATUACAO = A.ID

```

```

9      JOIN COLONIA C
10         ON C.PLANETA = A.PLANETA_CIVIL
11         AND C.TURNO_INICIAL <= A.TURNO_CIVIL
12         AND (C.TURNO_FINAL >= A.TURNO_CIVIL OR C.TURNO_FINAL IS NULL)
13         AND IMPERIO = :IMPERIO
14      RIGHT JOIN RECURSOS R
15         ON G.RECURSO = R.NOME
16 GROUP BY
17      R.NOME
18 ORDER BY
19      R.NOME
20 ;

```

Localização: */database/consultas.sql*

Neste trecho de código, cada recurso produzido por um império através das múltiplas construções e civis em todas as suas colônias, além da quantidade produzida, são selecionados. Para obter esses dados, a consulta utiliza a relação Colônia, para saber quais planetas são ou eram do império, atrelando-a à relação Atuação pelo planeta e turno da posse pelo império fornecido. Isso faz com que a produção seja contada mesmo em planetas que já não são mais colônias imperiais. As tabelas Gera\_recurso e Recursos também são utilizadas para poder atrelar a atuação a um recurso e seu nome. Esta última ocorre com uma junção direita, visto que é importante listar até mesmo os recursos que não foram gerados, com quantidade 0. O agrupamento e ordenação são feitos pelo nome do recurso, para conseguir somar a quantidade produzida de recurso.

A relevância desta consulta ao projeto ocorre em funcionalidades nas quais é importante obter estatísticas sobre o avanço e progresso do jogador. A quantidade total produzida é um número que pode agir psicologicamente em um usuário como motivação, além de motivo de orgulho e aumento de rivalidade saudável entre jogadores.

### 6.3. Quantidade de Civis Especializados de cada Espécie por Determinado Império até Dado Turno

```

1 SELECT
2     E.NOME AS ESPECIE,
3     NVL(SUM(A.QTD_TRABALHADORES), 0) AS CIVIS
4 FROM
5     ATUACAO A
6     JOIN COLONIA C
7         ON (IMPERIO = :IMPERIO AND A.TURNO.CONSTRUCAO <= :TURNO)
8         AND (C.PLANETA = A.PLANETA_CIVIL AND C.TURNO_INICIAL <= A.TURNO_CIVIL
9         AND (C.TURNO_FINAL >= A.TURNO_CIVIL OR C.TURNO_FINAL IS NULL))
10    RIGHT JOIN ESPECIE E
11        ON E.NOME = A.ESPECIE
12 GROUP BY
13     E.NOME
14 ORDER BY
15     E.NOME

```

16 ;

Localização: */database/consultas.sql*

Esta consulta devolve a quantidade de civis, agrupados por espécie, transformados em outras ocupações por um determinado império desde o começo do jogo até certo turno. Para obter tal dado, a tabela Atuação é unida à relação Colônia pelo planeta e turnos onde ocorrem. Também é unida à tabela Espécie por junção direita pela espécie que efetuou a atuação. Esse tipo específico de união ocorre por ser necessário obter espécies que não foram transformadas em nada pelo império, de forma que a sua soma dê zero. O agrupamento e ordenação ocorre pela espécie.

A importância desta estatística no jogo é, entre muitas outras, de demonstrar o avanço do jogador desde o começo do jogo no quesito especialização para cientista ou militar.

#### 6.4. Quantidade de Indivíduos de Cada Espécie Mortos em Batalhas Protagonizadas por dado Império

```

1 SELECT
2     E.NOME AS ESPECIE,
3     NVL(SUM(PM.QTD), 0) AS MORTOS
4 FROM
5     POVO.MORTO.BATALHA PM
6     JOIN BATALHA B
7         ON PM.ID_BATALHA = B.ID AND B.IMPERIO = :IMPERIO
8     RIGHT JOIN ESPECIE E
9         ON PM.ESPECIE = E.NOME
10    GROUP BY
11        E.NOME
12 ;
```

Localização: */database/consultas.sql*

Nesta seção, a quantidade de mortos em todas as batalha protagonizadas (de ataque) por um império específico até o atual turno são obtidos, agrupados por espécie. Para isso, a tabela Batalha é utilizada para definir todas nas quais um império participou. A cada uma delas, atrelam-se as tuplas da tabela Povo\_morto\_batalha para definir quantidade de mortos. Finalmente, é utilizada uma junção direita na tabela de espécies para poder obter até espécies com nenhum morto em batalhas do império, exibindo 0 mortos na saída. A ordenação também é feita pelo nome das espécies, igual à segunda consulta.

A importância desta estatística é demonstrar, como a anterior, o progresso do jogador ao longo da partida. Por isso, possui todos os pontos principais da segunda consulta. Também pode ser utilizada como indicativo de espécies das quais o jogador ainda não tirou proveito nem experimentou eliminar.

#### 6.5. Checagem se Dado Império Consegue Construir Dada Estrutura com seu Estoque Disponível em dado Turno

```

1 SELECT
2     MIN((NVL(E.QUANTIDADE, 0) - R.QTD)) AS PODE_CONSTRUIR
3 FROM
```



```

4      RECURSO_PARA_ESTRUTURA R
5      LEFT JOIN ESTOQUE E
6          ON R.RECURSO = E.RECURSO
7 WHERE
8      E.PLANETA = :PLANETA AND E.TURNO = :TURNO
9      AND R.ESTRUTURA = :ESTRUTURA
10 ;

```

Localização: */database/consultas.sql*

Esta consulta imprime um valor inteiro dependendo da habilidade de um império construir certa estrutura em certo turno em certo planeta. Caso o valor retornado seja menor que 0, é impossível efetuar a construção, pois os recursos são inferiores ao necessário. Já valores maiores ou iguais a 0 demonstram que o império possui a habilidade de construir. Para isso, a tabela *Recurso\_para\_estrutura* é unida por junção direita ao estoque. Esse tipo de junção é utilizado por ser importante obter materiais necessários para a estrutura, mas não existentes no estoque.

Tal informação é pivô na mecânica de construções do jogo por ser utilizada para verificação toda vez que o jogador quiser ou tentar efetuar uma construção.

#### 6.6. Listagem de planetas que já construíram todas as estruturas avançadas

```

1 WITH ESTRUTURAS_AVANCADAS AS (
2     SELECT
3         E.NOME
4     FROM
5         ESTRUTURA E
6     WHERE
7         E.TIPO = 'INDUSTRIA'
8         AND (E.NOME IN ( 'PEDREIRA TECNOLÓGICA', 'MINERADORA',
9             'FABRICA DE CHIP AUTOMÁTICA', 'FABRICA DE NAVE ROBOTIZADA',
10            'INDUSTRIA DE ARMAS ROBOTIZADA', 'FABRICA DE GERADOR'))
11 )
12 SELECT
13     P.NOME
14 FROM
15     PLANETA P
16 WHERE
17     NOT EXISTS ( (
18         SELECT
19             ES.NOME
20         FROM
21             ESTRUTURAS_AVANCADAS ES
22         ) MINUS (
23             SELECT
24                 C.ESTRUTURA
25             FROM
26                 CONSTRUCAO C

```

```

27         WHERE
28             C.PLANETA = P.NOME
29         ) )
30 ;

```

Localização: */database/consultas.sql*

Esta consulta imprime uma lista com o nome dos planetas que já construíram todas as estruturas consideradas avançadas (PEDREIRA TECNOLÓGICA, MINERADORA, FABRICA DE CHIP AUTOMÁTICA, FABRICA DE NAVE ROBOTIZADA, INDUSTRIA DE ARMAS ROBOTIZADA, FABRICA DE GERADOR). Essa consulta serve para descobrir os planetas que atingiram o nível tecnológico máximo em relação a estruturas utilizando divisão relacional.

### 6.7. Classificação dos Impérios por Estimativa de Potencial de Poder Total em Dado Turno

```

1  SELECT
2      C.IMPERIO ,
3      I.COR,
4      NVL(SUM(P.QTD_AGUA * P.ESTRUTURAS_MAX) , 0) AS PODER
5  FROM
6      PLANETA P
7      JOIN COLONIA C
8          ON P.NOME = C.PLANETA
9          AND C.TURNO_INICIAL <= :TURNO
10         AND (C.TURNO_FINAL >= :TURNO OR C.TURNO_FINAL IS NULL)
11      JOIN IMPERIO I
12          ON I.NOME = C.IMPERIO
13  GROUP BY
14      C.IMPERIO ,
15      I.COR
16  ORDER BY
17      NVL(SUM(P.QTD_AGUA * P.ESTRUTURAS_MAX) , 0) DESC
18 ;

```

Localização: */database/consultas.sql*

Para demonstrar o potencial de crescimento dos impérios somente com as colônias atuais, é criada uma estimativa de potencial de poder por colônia. Este valor representa uma estimativa do poder máximo que pode ser extraído daquele determinado planeta baseado em suas características. A soma disto para todas as colônias resulta na estimativa de potencial de poder total de um império. Para obter a classificação dos impérios a partir disto, as tabelas Planeta, Colônia e Império foram unidas para obter as características de todos os planetas pertencentes a um império no turno atual. A partir destas características, pode-se classificar cada império em ordem decrescente de potencial.

Tal operação é muito importante para obter-se visão geral de como está o progresso de cada império em comparação aos outros. Além de fomentar competição saudável, poder ver sua classificação também ajuda o jogador a saber em quais áreas precisa investir mais. Por exemplo, se está em uma ótima classificação em potencial de poder total, mas perde batalhas constantemente

e não consegue manter seus planetas, talvez isso signifique que o jogador possui planetas com muito potencial, mas não o desenvolve. Isso faz com que o jogador busque desenvolver melhor as estruturas de seus planetas para extrair o potencial máximo, evitando só ficar obtendo mais e mais planetas sem focar em suas prosperidades.

#### 6.8. Estimativa de Potencial de Poder de Cada Colônia de cada Império em um dado Turno

```

1 SELECT
2     I.NOME,
3     I.COR,
4     P.NOME,
5     NVL(P.QTD_AGUA * P.ESTRUTURAS_MAX, 0) AS PODER
6 FROM
7     PLANETA P
8     JOIN COLONIA C
9         ON P.NOME = C.PLANETA
10        AND C.TURNO_INICIAL <= :TURNO
11        AND (C.TURNO_FINAL >= :TURNO OR C.TURNO_FINAL IS NULL)
12     JOIN IMPERIO I
13         ON I.NOME = C.IMPERIO
14 ORDER BY
15     I.NOME,
16     NVL(P.QTD_AGUA * P.ESTRUTURAS_MAX, 0) DESC
17 ;

```

Localização: */database/consultas.sql*

Para obter estes dados, as tabelas Planeta, Colônia e Império foram unidas para obter as características de todos os planetas que são colônia de um determinado império em um dado turno. A partir destas características, calcula-se a estimativa do potencial de poder para cada colônia e classifica-as por esta grandeza.

A consulta em questão é responsável por fornecer ao jogador uma visão mais individualista da estimativa de potencial de poder de cada uma das suas colônias, para entregar informações sobre quais os seus planetas com maior potencial de crescimento. Assim, o usuário pode ter uma noção melhor de quais planetas deve defender ou desenvolver com maior ênfase, auxiliando no seu progresso.

## 7. DESCRIÇÃO DA IMPLEMENTAÇÃO DO PROJETO

### 7.1. Organização e Arquitetura do Sistema

Para a prototipagem do jogo, o grupo optou por utilizar a linguagem Python devido a sua maior praticidade de implementação, alto nível e disponibilidade de bibliotecas. A ausência de necessidade técnica por mais desempenho de código ou implementações gráficas mais requintadas do que a interface de linha de comando criada fizeram com que essa linguagem fosse a melhor opção.

A organização principal do código é composta pelos seguintes diretórios e arquivos:

---

Diretório <i>database/</i> .....	Códigos Fonte do Banco de Dados
Diretório <i>documentation/</i> .....	MR e DER do Banco de Dados
Diretório <i>project/</i> .....	Implementação do Projeto em Python
Arquivos <i>README.MD, LICENSE</i> .....	README e Licença do projeto

---

No contexto das operações principais do SQL, o diretório *database* é composto pelos seguintes arquivos:

---

Arquivo <i>consultas.sql</i> .....	Consultas ao BD
Arquivo <i>dados.sql</i> .....	Scripts de Preenchimento Inicial do BD
Arquivo <i>esquema.sql</i> .....	Criação das Tabelas e Estrutura do BD
Arquivo <i>limpador.sql</i> .....	Script de Eliminação do BD Completo

---

No contexto das operações principais do Projeto, o diretório *project* é composto pelos seguintes arquivos e seus conteúdos:

---

Diretório <i>controller/</i> .....	Impressões da Interface CLI
Diretório <i>models/</i> .....	Objetos Utilizados no Código
Diretório <i>view/</i> .....	Tela Inicial do Jogo
Arquivo <i>main.py</i> .....	Script Inicializador do Jogo

---

### 7.2. Explicação sobre a Aplicação

#### 7.2.1 Visão Geral

Resumidamente, o projeto se trata de um jogo simples no qual de 1 a 3 usuários inserem o nome e a cor de seu Império, em seguida escolhendo um planeta para conquistar. Este passará a ser uma colônia de seu Império. Cada jogador escolhe um planeta disponível para colonização 3 vezes, e no final é exibido um ranking dos Impérios que pegaram as colônias com maior potencial.

O jogo foi feito utilizando a linguagem de programação Python e o Sistema Gerenciador de Bancos de Dados da Oracle.

Nas seções seguintes discorrerá-se alguns detalhes mais técnicos da implementação, com foco principal naquilo que tange a parte de Base de Dados, não necessariamente na lógica do código

como um todo. Tais detalhes serão um pouco mais abordados na apresentação do vigente trabalho e são mostrados também no código-fonte, entregues anexos ao relatório.

### 7.2.2 Detalhes técnicos da implementação

Foram feitas todas as tratativas de erros, tanto ao nível de captura de entrada de dados pelo usuário quanto ao nível de captura de exceções no código. Exemplos de possíveis erros previstos e tratados são inserção de tipos não compatíveis com variáveis no código ou restrição de atributo da base de dados.

O código foi feito numa estrutura baseada no Modelo MVC, ou seja, Model, View e Controller. Existe a pasta "models", a pasta "view" e a pasta "controller", na primeira estão as classes que representam objetos tangíveis na aplicação, na segunda está a classe responsável por toda a geração da interface de usuário (no caso deste projeto é toda na linha de comando do console) e por fim, na terceira está a classe do motor principal do jogo além da classe responsável por fazer a configuração básica da base de dados da Oracle.

### 7.2.3 Controle Transacional do Código

Nessa aplicação foi feito um controle transacional bem simples, considerando apenas uma máquina por vez rodando o código do jogo e acessando a base de dados, ou seja, as transações são todas seriais e não há concorrência. O controle de transação de todo o código baseia-se no uso de "try catch". Se um comando SQL foi executado corretamente, dá-se o "commit" no código, caso uma exceção seja capturada, dá-se o "rollback".

Segue abaixo um exemplo, que se trata da função insere uma colônia. Todas as transações são realizadas de maneira análoga.

```

1 def insertColony ( self , empire , planet , turn , connection ) :
2     cursor = connection . cursor ()
3
4     error = False
5     try :
6         cursor . execute ( """
7             INSERT INTO COLONIA
8             (IMPERIO, PLANETA, TURNO_INICIAL, TURNO_FINAL)
9             VALUES (:empireSQL, :planetSQL, :turnSQL, :turnEndSQL)
10            """ ,
11            empireSQL = empire , planetSQL = planet , turnSQL = turn , turnEndSQL = 3 )
12        connection . commit ()
13
14    except oracledb . DatabaseError as e :
15        self . view . printSystem ( " Erro ocorreu : { } " . format ( e ) )
16        error = True
17        connection . rollback ()
18
19    finally :
20        cursor . close ()
21        self . colonies . append ( Colony ( empire , planet , turn ) )

```

22

`return error`

### 7.2.4 Comandos SQL utilizados na aplicação

A aplicação contém vários comandos SQL que seguem as boas práticas de utilização destes. A conexão com a base de dados é feita no Python utilizando-se da biblioteca "oracledb".

A conexão é feita no início do código e fechada no final. Para cada execução de comando SQL abre-se um cursor e depois fecha-se, conforme mostrado na seção sobre o Controle Transacional.

Foi feita uma parte de "configuração da base de dados", com a conexão, a desconexão, a remoção de outros esquemas que possam estar na Base de Dados remanescentes de outras práticas, criação dos esquemas necessários para a aplicação (simplificação do modelo completo), e inserção de alguns dados básicos para o início da aplicação.

A parte da jogatina em si contém vários comandos para a inserção de dados dos Impérios(jogadores), inserção de dados no esquema de Colônias, o que refere-se a colonização de um planeta por um Império e consultas para listar os jogadores, os planetas disponíveis para colonização e a quantidade de pontos dos Impérios, definida como a soma de "Estimativa de Potencial de Poder" que é a multiplicação da quantidade de água de um planeta pela quantidade de estruturas máximas que aquele planeta pode ter. Abaixo estão expostas algumas das funções do código que usam comandos SQL. Não foram mostradas todas, pois essa seção ficaria desnecessariamente grande. Além disso, todo o código-fonte segue juntamente com a entrega desse relatório, portanto para mais detalhes basta olhar no código, que está devidamente modularizado, com funções e variáveis nomeadas adequadamente e bem comentado.

Exemplo de inserção: a inserção de Império, com parâmetros passados pelo usuário:

```

1      def insertEmpire ( self , name , color , connection ) :
2          cursor = connection . cursor ( )
3
4          error = False
5          try :
6              cursor . execute ( """
7                  INSERT INTO IMPERIO (NOME, COR)
8                  VALUES (:nameSQL, :colorSQL)
9                  """ ,
10                 nameSQL = name , colorSQL = color )
11
12             connection . commit ( )
13
14         except oracledb . DatabaseError as e :
15             self . view . printSystem ( "Erro : {e} ". format ( e ) )
16             error = True
17             connection . rollback ( )
18
19         finally :
20             self . empires . append ( Empire ( name , color ) )
21             cursor . close ( )
22             return error

```

Exemplo de consulta abaixo, nesse caso a consulta de mostrar o ranking dos Impérios, provavelmente a consulta mais complexa da aplicação, essa consulta consiste em mostrar os Impérios, cada um com sua cor e sua pontuação definida pela soma da Estimativa de Potencial de Poder, que é a multiplicação da quantidade de água de um planeta vezes a quantidade máxima de estruturas que cabem naquele planeta, de todos seus planetas, agrupado pelo Império e ordenado pelo Poder do planeta.

```

1 def printRanking(self, connection):
2     cursor = connection.cursor()
3     self.view.clear()
4     ranking = 0
5     try:
6         cursor.execute("""SELECT C.IMPERIO, I.COR,
7             NVL(SUM(P.QTD_AGUA *
8             P.ESTRUTURAS_MAX), 0) AS PODER FROM PLANETA P
9             JOIN COLONIA C ON P.NOME = C.PLANETA
10            AND C.TURNO_INICIAL <= :TURNO
11            AND (C.TURNO_FINAL >= :TURNO
12            OR C.TURNO_FINAL IS NULL)
13            JOIN IMPERIO I ON I.NOME = C.IMPERIO
14            GROUP BY C.IMPERIO, I.COR
15            ORDER BY NVL
16            (SUM(P.QTD_AGUA * P.ESTRUTURAS_MAX), 0)
17            DESC""", TURNO = 3)
18        rows = cursor.fetchall()
19        if not rows:
20            self.view.printSystem("Nenhum retorno em Imperios.")
21        else:
22            for name, color, power in rows:
23                ranking+=1
24                self.view.printColorMessage("""{} - Imperio:{}
25                Poder Total:{}""".format(ranking,name,power), color)
26    except oracledb.DatabaseError as e:
27        self.view.printSystem("Erro: {}".format(e))
28        connection.rollback()
29
30    finally:
31        cursor.close()

```

### 7.2.5 Para rodar o projeto

O projeto foi feito em Python, e usando o Sistema Gerenciador de Banco de Dados da Oracle, então para rodar o código-fonte é necessária a devida instalação desses softwares no computador.

Além disso, o uso de algumas bibliotecas exigem instalação de pacotes, como a biblioteca

oracledb, e a dotenv, a primeira utilizada para se conectar a base de dados por meio do Python e a segunda para inserir variáveis de ambiente no código, nesse caso o login e senha da conexão na base de dados. Essas bibliotecas são instaladas usando-se o comando "pip install oracledb" e "pip install python-dotenv" no console.



## 8. CONCLUSÃO

Na visão geral do projeto, é possível inferir que houve significativo aprendizado em todas as partes da aplicação dos conceitos de sala. Sobre isso, a maior dificuldade ocorreu devido à complexidade e dimensões do modelo de jogo criado. A visão inicial do grupo sobre a abordagem de criação do jogo, após diálogos com o monitor da disciplina, foi definida como muito superficial para a proposta do trabalho. Como resultado, houve um contrapeso da equipe em prol de criar algo significativamente mais complexo, o que agiu como faca de dois gumes. Se por um lado tal criatividade tornou possível uma visão mais complexa da disciplina, por meio do tratamento de diversos casos antes não imaginados no planejamento de banco de dados, também foi, junto da documentação limitada em relação ao escopo do jogo, causa de performance pobre na primeira entrega. Especificamente, a utilização de histórico de turnos em muitas entidades do trabalho tornou diversos relacionamentos complexos, e, junto da extensiva jogabilidade sendo fomentada nas mentes dos integrantes, fez com que uma mudança de abordagem fosse necessária. É pensando nisso que, para a segunda entrega, o time se dedicou em criar uma documentação digna de suas imaginações, inserindo todas as intrincadas e específicas fórmulas e mecânicas de jogabilidade desejadas e explicitando seus objetivos com o MER e MR, ao custo de múltiplas horas de dedicação. Para uma disciplina de somente 2 créditos trabalho, acredita-se que o requerimento total ocupou significativamente mais horas do que a quantia declarada na ementa.

No que tange o aprendizado, acredita-se que todas as entregas foram mais que suficientes para aplicação dos conceitos vistos em sala, indo, na verdade, além. Explorou-se, além dos conceitos de Bases de Dados, todas as ferramentas que giram em torno do desenvolvimento deste tipo de sistema. Apesar da impossibilidade de tempo para criar algo mais complexo graficamente, o grupo também aprendeu mais sobre desenvolvimento geral de aplicações em Python. Para futuras turmas, é desejada maior noção de complexidade de projeto antes do desenvolvimento deste.

Há grande satisfação por parte da equipe em finalmente entregar o projeto completo, pois foi possível prover um resultado final do qual há orgulho, ainda mais tendo em vista todo o desenvolvimento e dedicação aplicados em cima da proposta inicial. O time agradece à professora e monitor da disciplina pelo apoio e ensinamentos.

## 9. REFERÊNCIAS

- **ELMASRI, R; NAVATHE, S.B.**, Sistemas de Banco de Dados, Addison Wesley.
- **Silberschatz, A. et al.**, Sistema de Banco de Dados, MCGrow-Hill.