

Nome: Antônio Gustavo Muniz
Nome: Gabriel Bueno
Nome: Henrique Vital
Nome: João Vitor Dias
Nome: Weverson da Silva

RA: 22.119.001-0
RA: 22.119.077-0
RA: 22.119.078-8
RA: 22.119.006-9
RA: 22.119.004-4

Etapa 2

1. Descrição do Sistema

1.1. Objetivo

O objetivo do sistema é modelar um banco de dados para um *massively multiplayer online role-playing game* (MMORPG), um jogo em que é possível ser quem você quiser e com os poderes que você quiser. O foco desse jogo é o combate entre jogadores ou entre inimigos do jogo, mas, para isso, o jogador precisa melhorar seus equipamentos, se aliar a algum clã e conversar com outros jogadores. Uma base de dados de um MMORPG real é incomparavelmente maior que o apresentado aqui, mas por motivos didáticos, focaremos apenas no personagem e as informações que condizem com ele, como atributos, itens e habilidades.

1.2. Funcionalidades

O sistema tem como função inicial o cadastro de novos personagens. Cada *Personagem* possui a sua raça e a sua classe. Com base nisso, os atributos do personagem, a árvore de habilidades e os itens iniciais serão fornecidos. Todos os iniciantes começam com as mesmas características, que podem ser melhorados durante o jogo.

Personagens podem interagir entre si através de combate, troca de itens ou se comunicando mesmo. No combate, o personagem consegue ganhar itens do personagem inimigo derrotado. Na troca de itens, o personagem consegue dar e receber itens de outro personagem. Cada personagem pode ser dividido entre jogável e não-jogável (NPC). Um personagem jogável é aquele que pode ser controlado por uma pessoa real e um não-jogável é aquele controlado pelo computador. Cada personagem não-jogável tem uma certa afinidade com um personagem jogável, pode ser aliado ou inimigo de um jogador e possui uma determinada quantidade de pontos de experiência que podem ser obtidos por um personagem jogável ao ser morto por ele.

Além disso, o personagem jogável consegue checar os itens que

ele carrega consigo e ver se ainda tem espaço para algo. Cada item apresenta informações diferentes. Os equipamentos (como armadura e espada) mostram, dentre outras informações, a durabilidade e o bônus de ataque/defesa, mas também alertam as aptidões que o equipamento exige para ser utilizado, por exemplo: Constituição maior que 15 para utilizar a espada gélida. Itens como Moeda e *Variedades* não podem ser usados em batalha, mas o jogador pode armazená-las para futuras trocas ou construções.

Por fim, o sistema disponibilizará os dados mais comuns entre os jogadores. Por exemplo: classes, raças, itens e habilidades mais escolhidos, assim como vilões mais combatidos. Servindo assim para ajudar os jogadores iniciantes a se divertirem mais, não se preocupando com as *builds* (atributos, itens, habilidades, entre outros) e com o que comprar.

2. Modelagem Entidade Relacionamento

É possível ver os dois modelos entidade relacionamento em uma melhor qualidade na pasta "Modelo Entidade Relacionamento/" ou clicando [aqui](#).

2.1. MER anterior VS MER atual

Durante a criação da Modelo Relacionamento, nós percebemos algumas inconsistências no MER anterior.

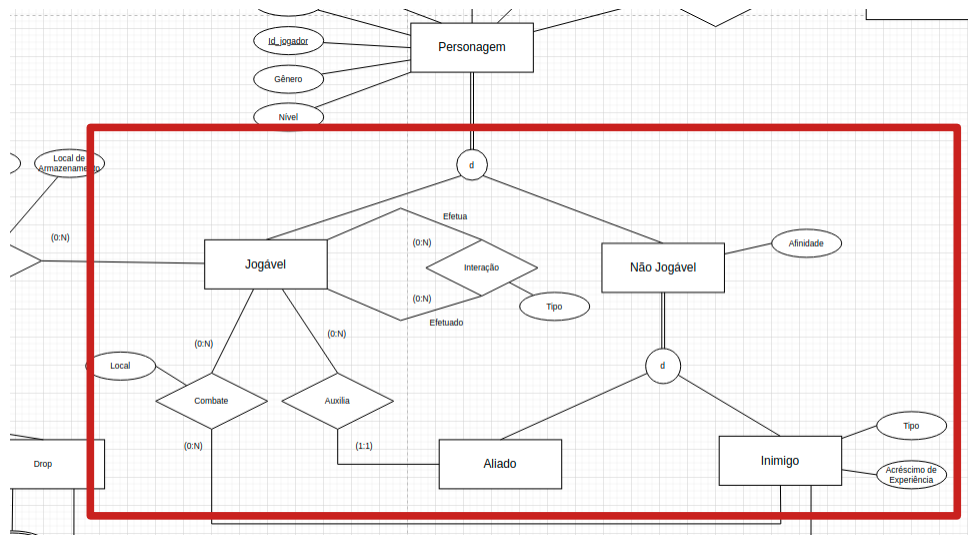


Figura 1: Disposição das entidades Jogável e Não-jogável

Anteriormente, a relação de interação era só entre entidades *Jogável*, ou seja, só jogadores poderiam interagir entre si na troca de itens e na comunicação. Além disso, a única relação entre *Jogável* e *Não Jogável* era de combate, o que tornava o rpg limitado engessado. Porém, na nossa nova modelagem, nós decidimos mudar um pouco essa disposição.

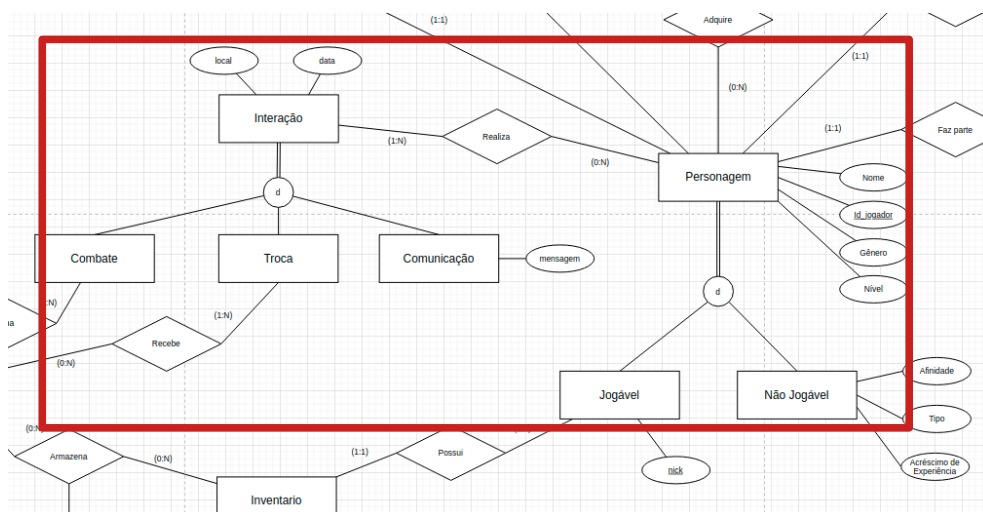


Figura 2: Disposição das entidades Jogável e Não Jogável no novo MER

Nós removemos as entidades *Aliado* e *Inimigo*, pois o que define um aliado e um inimigo é a afinidade dele com o jogador e as relações que estão contidas na entidade *Personagem*. Essa mudança torna a relação entre *Jogáveis* e *Não jogáveis* muito mais clara. Além disso, nós convertimos a relação de interação em uma entidade que está ligada ao personagem. Desse modo, conseguimos guardar as interações entre diferentes usuários e tornar elas mais ricas.

Além disso, anteriormente, a única relação que a entidade *Item* possuía era com a entidade *Jogável*, que indicava que o *Jogável* podia guardar um *Item*.

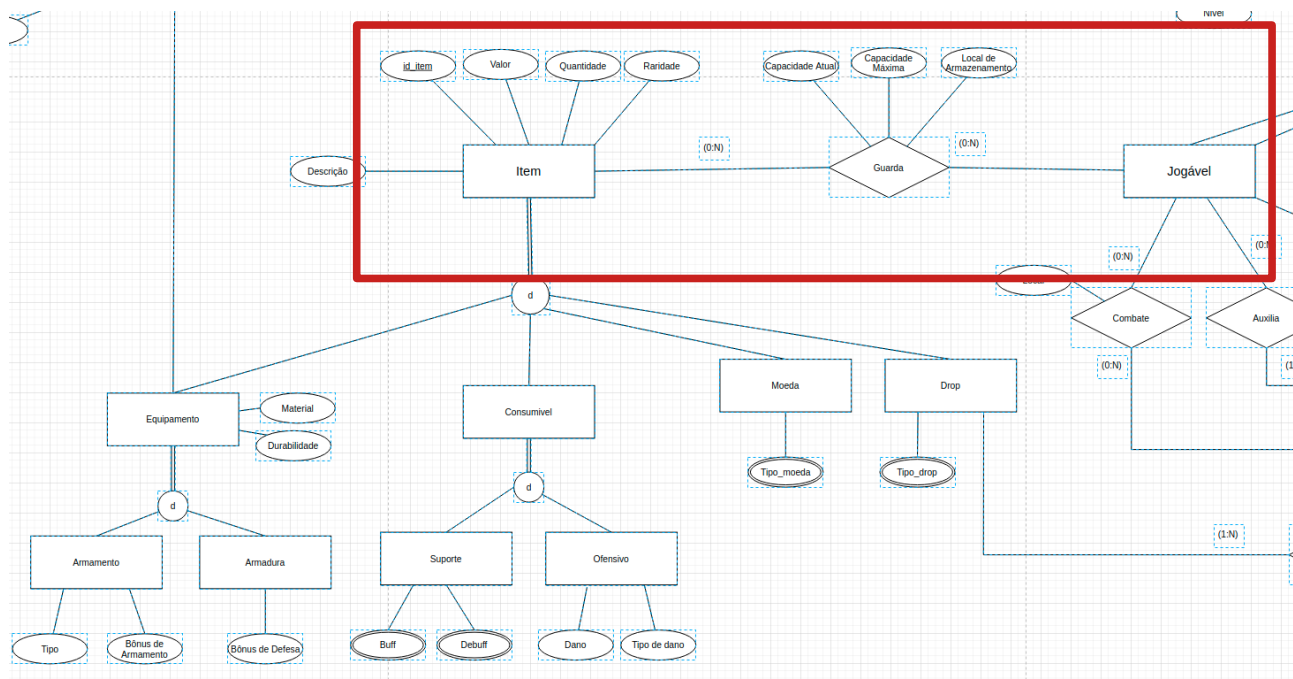


Figure 3: Relação prévia entre entidade Jogável e Item

No novo MER, *Jogável* se relaciona com *Item* por intermédio da entidade *Inventário*, explicitando ainda mais que o personagem jogável guarda um item através do seu inventário. Além disso, *Item* agora se relaciona com as entidades *Combate* e *Troca*, deixando claro que é possível que um personagem ganhe itens através da interação com outros personagens.

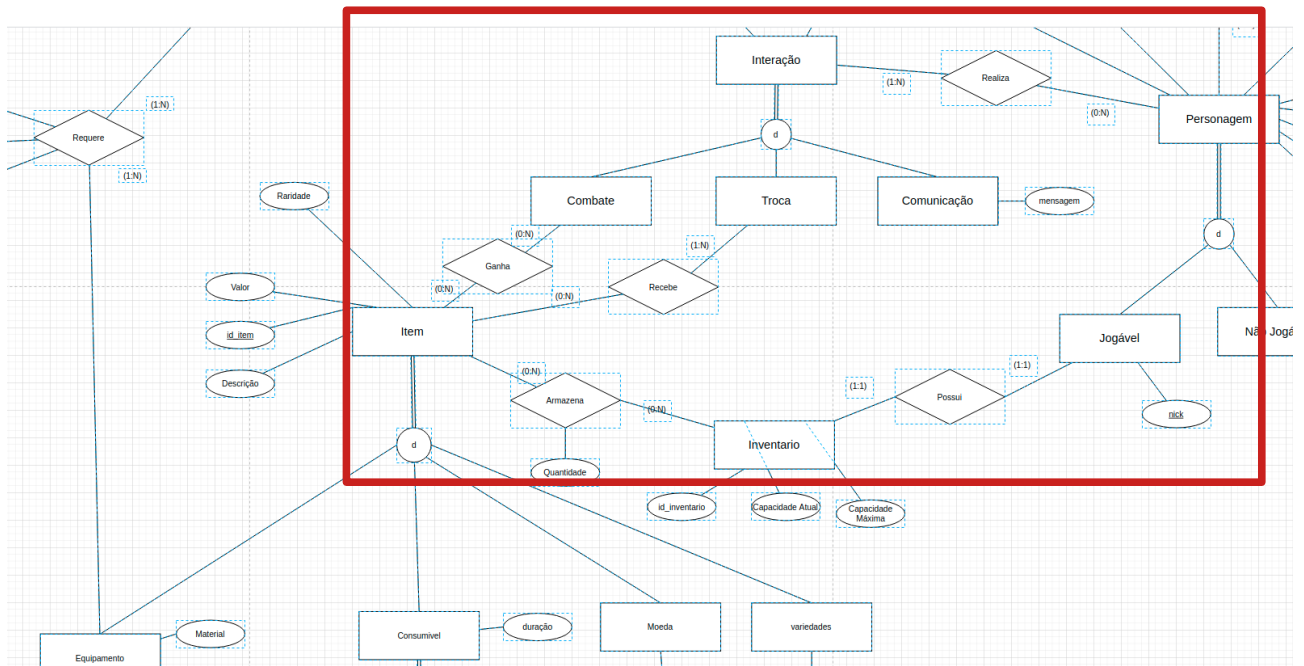


Figure 4: Novas relações envolvendo Item, Personagem, Inventário e Interação

Finalmente, no novo MER, nós removemos a relação entre as entidades *Atributo* e *Aptidões* a favor da relação entre *Aptidões* e *Personagem*. Ou seja, no final das contas, *Aptidões* está conectada tanto com *Equipamento*, evidenciando os requisitos mínimos para a empunhadura daquele equipamento, quanto com *Personagem*, definindo as características principais dele.

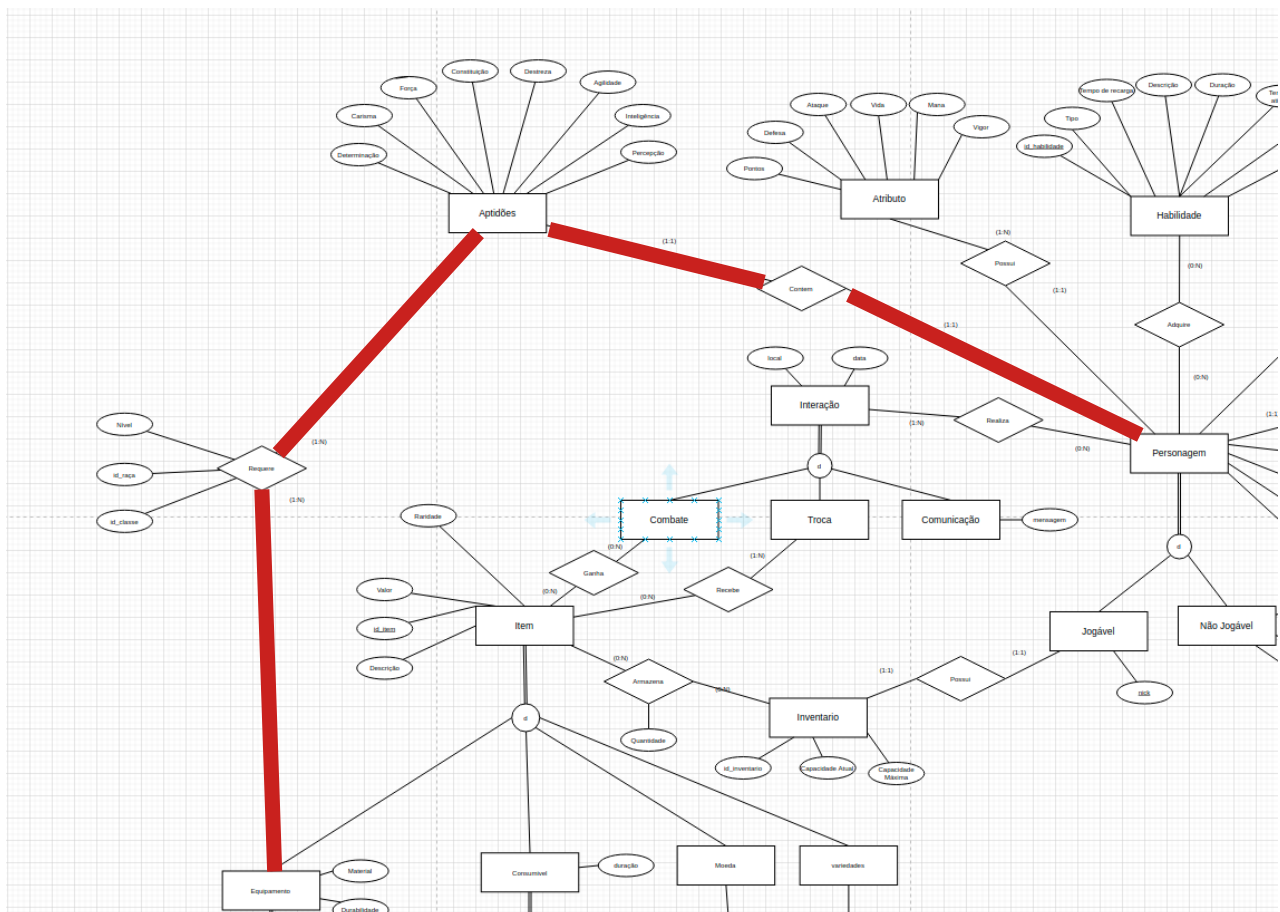


Figure 5: Relação das entidades Personagem e Equipamento com Aptidões

3. Modelo Relacional

É possível ver os dois modelos entidade relacionamento em uma melhor qualidade na pasta "Modelo Relacionamento/" ou clicando [aqui](#).

No modelo relacional, como a relação entre *Personagem* e *Habilidade* é de $N \rightarrow N$, é necessita-se criar uma tabela intermediária denominada *Habilidade_jogador*, que armazena as combinações entre as entidades *Personagem* e *Habilidade*, juntamente com o nível da habilidade.

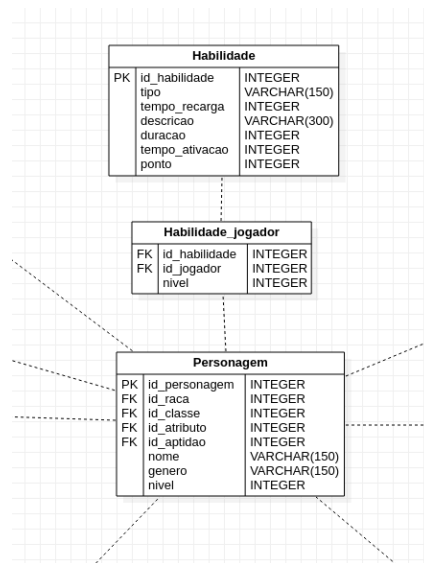


Figure 6: Representação no MR da relação entre *Personagem* e *Habilidade*

Prosseguindo, um dos problemas enfrentados durante a criação do modelo relacionamento foi a disjunção sem sentido de *Não-jogável* em *Aliado* e *Inimigo*, como dito na seção 2.1. Seguindo o novo MER, elas foram removidas e estão implícitas dentro das interações.

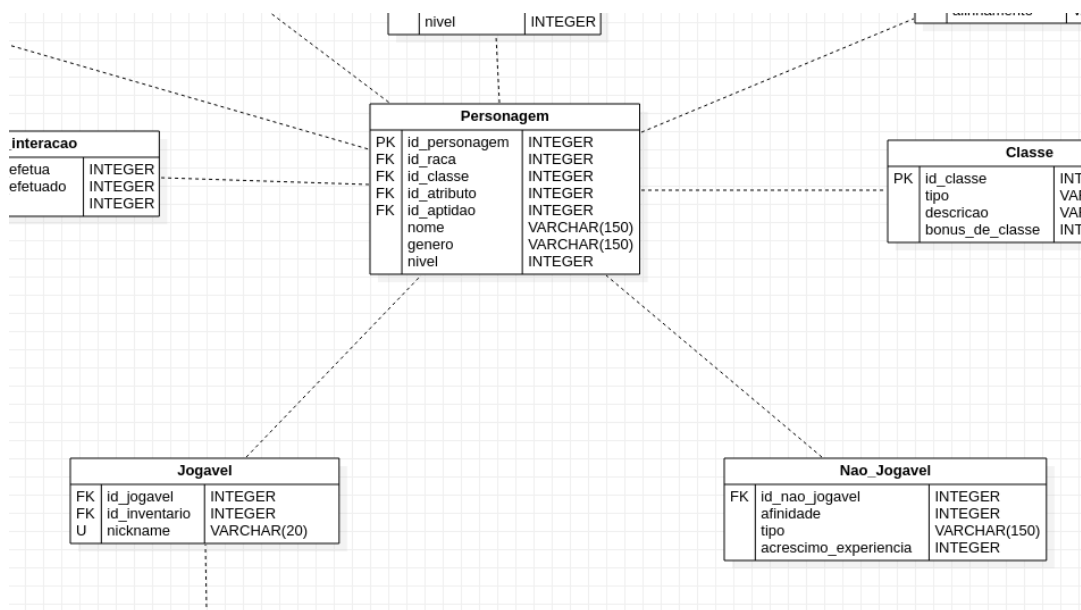


Figure 7: Relação entre Jogável e Não-jogável

Outra parte interessante do nosso modelo de relacionamento é a interação entre personagens. Ela que desencadeou a mudança no MER anterior. Basicamente, há uma relação $N \rightarrow N$ entre *Personagem* e *Interação*, conseqüentemente, gerando uma tabela intermediária denominada *Realiza_interacao*. Além disso, criamos três tipos de interação, sendo elas *Combate*, *Troca* e *Comunicação*, com *Combate* e *Troca* se relacionando com *Item*. Essa relação é $N \rightarrow N$, portanto houve a necessidade de uma tabela intermediária para essas duas relações.

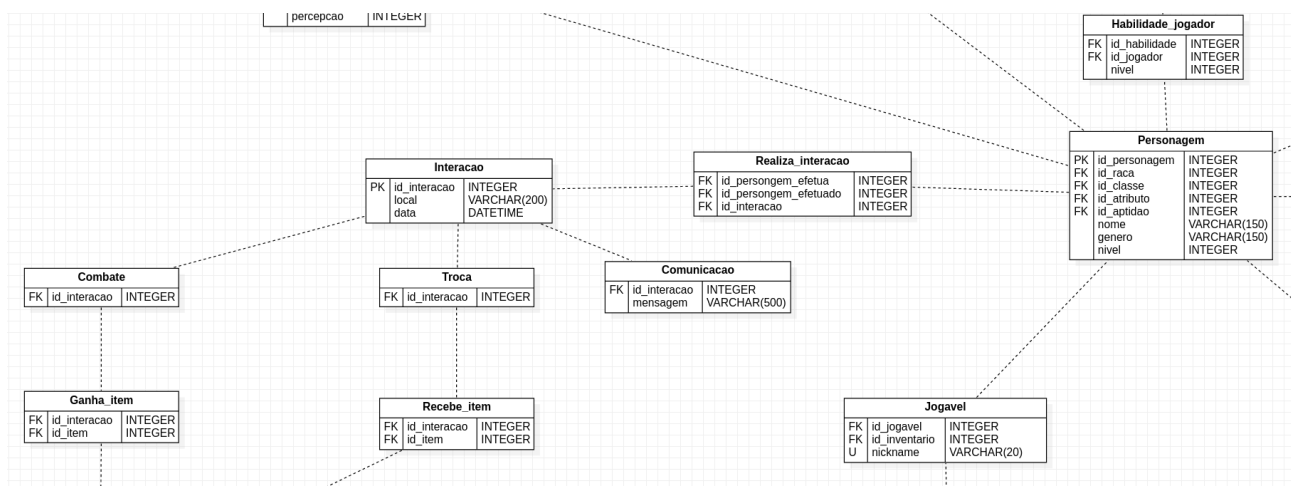


Figure 8: Fluxo de Interação entre Personagens

Outra parcela curiosa do MR é a relação de armazenamento de *Item* por parte de *Jogável*, que demanda que a entidade *Inventario* tenha um relacionamento $N \rightarrow N$ com *Item*. Isso gera uma tabela intermediária chamada *Armazena*, que guarda chaves estrangeiras de *Item* e *Inventario* e a quantidade de itens armazenados.

Além disso, temos também as entidades *Armamento* e *Armadura* que

possuem um relacionamento $N \rightarrow N$ com *Aptidao* e, portanto, geram uma tabela intermediária *Equipamento_requer_aptidao*. Como a relação de personagem com *Aptidão* é de $1 \rightarrow N$, só é necessário uma chave estrangeira *id_aptidao* em *Personagem*.

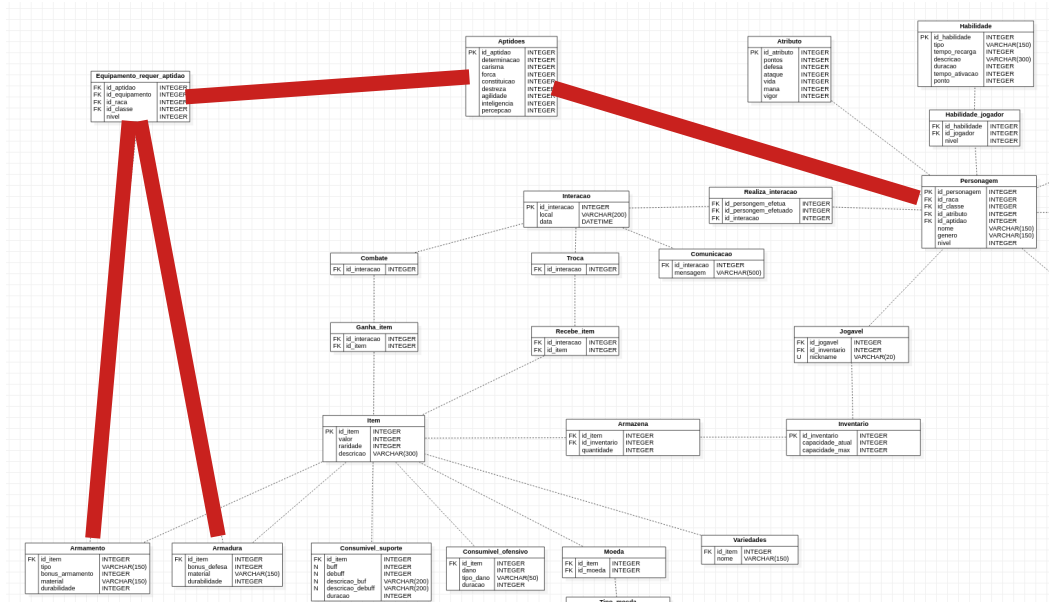


Figure 10: Modelagem Relacionamento de Equipamento com Aptidao e Personagem com Aptidao

Por fim, temos a disjunção aninhada em *Item*, que foi um dos nossos maiores desafios, pois, inicialmente, passávamos as chaves estrangeiras de *Item* como chaves estrangeiras de outras entidades. Esse tipo de operação não é suportado pelo Postgres e, conseqüentemente, tivemos que mudar nossa abordagem nessa parte da modelagem.

Nós decidimos subdividir *Item* em 6 tabelas, *Armamento*, *Armadura*, *Consumivel_suporte*, *Consumivel_ofensivo*, *Moeda* e *Variedades*. Isso ocorreu porque decidimos deletar as superclasses de *Armamento* e *Armadura* e de *Consumivel_suporte* e *Consumivel_ofensivo*. Dessa forma, mesmo algumas tabelas contendo valores que podem ser nulos, no fim das contas, conseguimos fazer a disjunção de *Item*.

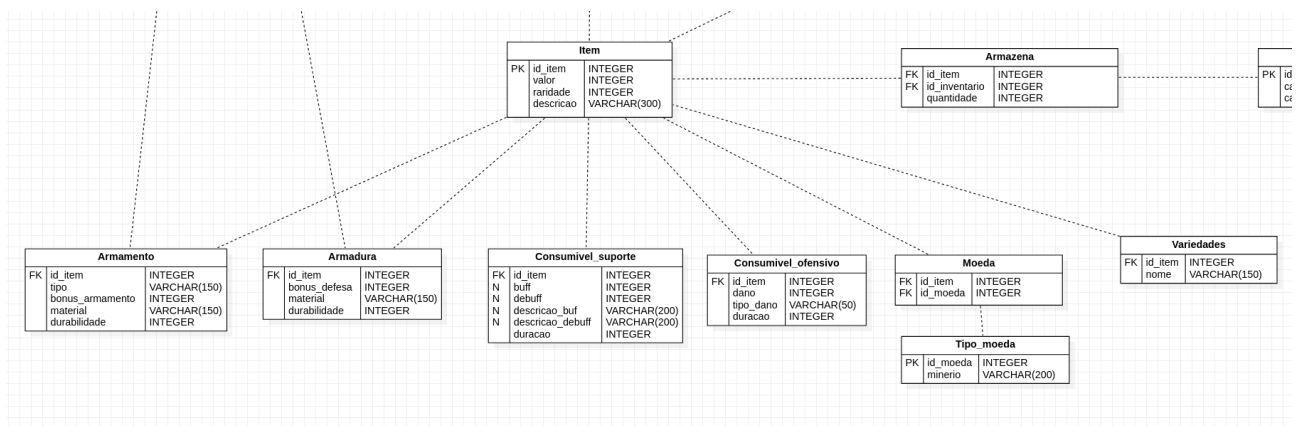


Figure 11: Disjunção de Item

4. Queries

É possível acessar todas as queries de criação, inserção e as queries especiais solicitadas clicando [aqui](#)

4.1. Query Especial

Encontrar qual o Personagem que mais combateu (com inimigo) e apresentar um profile do mesmo, considerando:

1. Seu nome
2. Qual seu equipamento mais utilizado
3. Qual o tempo médio de recarga
4. Qual é a sua maior aptidão
5. Seu nível de defesa e ataque"

```
SELECT id_personagem, nome, biggest_apt.aptidao, biggest_apt.valor_aptidao, atr.ataque, atr.defesa, AVG(tempo_recarga) as media_recarga,
max_quant_item.item_desc, max_quant_item.quant
FROM personagem
JOIN atributo atr ON atr.id_atributo = personagem.id_atributo
JOIN habilidade_jogador hj ON hj.id_jogador = personagem.id_personagem
JOIN habilidade h ON h.id_habilidade = hj.id_habilidade
JOIN (
    SELECT item.descricao as item_desc, MAX(arm.quantidade) as quant, jg.id_jogavel as id_jg
    FROM item
    JOIN armazena arm ON arm.id_item = item.id_item
    JOIN inventario inv ON inv.id_inventario = arm.id_inventario
    JOIN jogavel jg ON jg.id_inventario = inv.id_inventario
    GROUP BY item.descricao, jg.id_jogavel ORDER BY quant DESC LIMIT 1
) max_quant_item ON max_quant_item.id_jg = id_personagem
JOIN (
    SELECT
    CASE GREATEST(determinacao, carisma, forca, constituicao, destreza, agilidade, inteligencia, percepcao)
    WHEN determinacao THEN 'determinacao'
    WHEN carisma THEN 'carisma'
    WHEN forca THEN 'forca'
    WHEN constituicao THEN 'constituicao'
    WHEN destreza THEN 'destreza'
    WHEN agilidade THEN 'agilidade'
    WHEN inteligencia THEN 'inteligencia'
    WHEN percepcao THEN 'percepcao'
    END AS aptidao,
    GREATEST(determinacao, carisma, forca, constituicao, destreza, agilidade, inteligencia, percepcao) AS valor_aptidao,
    id_aptidao
    FROM aptidoes
) biggest_apt ON biggest_apt.id_aptidao = personagem.id_aptidao
WHERE id_personagem =
(
    SELECT id_perso as maiores_combatentes FROM
    (
    SELECT realiza_interacao.id_persongem_efetuado as id_perso
    FROM combate, realiza_interacao
    WHERE combate.id_interacao = realiza_interacao.id_interacao
    UNION ALL
    SELECT realiza_interacao.id_persongem_efetua
    FROM combate, realiza_interacao
    WHERE combate.id_interacao = realiza_interacao.id_interacao
    )
    AS maior_combatente
    GROUP BY maiores_combatentes ORDER BY COUNT(*) DESC LIMIT 1
)
GROUP BY id_personagem, biggest_apt.aptidao, biggest_apt.valor_aptidao, atr.ataque, atr.defesa, max_quant_item.item_desc, max_quant_item.quant
```

5. Definições

Algumas definições das entidades e atributos

Entidade Raça: A espécie do personagem. Exemplo: Anão, Elfo, Humano, Ogro

Alinhamento: Linha de ética e moral que guia as atitudes de um personagem. Exemplo: benfeitor, malfeitor, rebelde, espírito livre.

1. **Entidade Classe:** A profissão do personagem. Exemplo: Atirador, Lutador, Feiticeiro, Mago, Assassino, Comerciante, Ferreiro.
2. **Entidade Atributo:** Atributos mais impactados em batalhas.
 - **Vigor:** Semelhante a estamina ou resistência física. Com ele, é possível correr por mais tempo ou permanecer embaixo d'água por períodos prolongados.
3. **Entidade Aptidões:** Atributos que nascem com o personagem, mas podem ser melhorados com o decorrer do jogo
 - **Agilidade:** Com alta agilidade, o personagem tem mais chances de desviar de ataques e escapar de armadilhas.
 - **Carisma:** Com alto carisma, as pessoas estão mais propensas a gostar de você.
 - **Constituição:** Com alta constituição, atributos como vigor, saúde e condição físicas são melhoradas.
 - **Destreza:** Com alta destreza, o personagem lidar melhor com armas, ferramentas e instrumentos delicados.
 - **Determinação:** Com alta determinação, o personagem tem mais resistência à tortura psicológica, controle de mente entre outros.
 - **Força:** Com alta força, o personagem tem mais força física, pode arremessar e carregar objetos mais pesados.
 - **Inteligência:** Com alta inteligência, o personagem é dificilmente enganado, está mais apto a compreender o que está a sua volta e descobrir coisas novas.
 - **Percepção:** Com alta percepção, o personagem consegue observar detalhes importantes no ambiente, como detectar armadilhas e encontrar tesouros.
 - **Pontos:** Servem para melhoras os atributos listados acima e outros citados na entidade
4. **Entidade Consumível:** São equipamentos utilizáveis do personagem:
 - **Entidade Suporte:** Itens que apenas influenciam o personagem. Exemplo: poções de cura, regeneração de mana entre outros.
 - **Entidade Ofensiva:** Itens de ataque, como granadas, poções e veneno.
 - **Tipo de Dano:** Dano Mágico, Dano Físico e Dano Elemental, Dano Venenoso, Dano Psicológico.
5. **Atributo Afinidade:** Pertencente da entidade *Não Jogável*, define o quanto o personagem gosta de você. Por exemplo, um personagem com afinidade = -8 não gosta de você, mas um com afinidade = 0 está neutro em relação a você, e escolherá um lado dependendo das decisões tomadas por você.
6. **Atributos de Tipo:**

- **Na entidade *Variedades*:** O tipo de item recebido ao derrotar um inimigo. Exemplo: Garra de dragão, cabeça de caveira e coração congelado.
 - **Na entidade *Moeda*:** Tipos de moedas, como ouro, bronze, ferro, prata.
7. **Atributo *Bônus*:** Aumento de cada aptidão dos atributos da entidade *Atributo*
 8. **Relação *Interação*:** A interação entre os jogadores, que pode variar de um troca de itens e um auxílio em batalhas, a uma batalha entre os próprios jogadores.
 9. **Relação *Requiere*:** Para que certo equipamento seja utilizado são necessários certos níveis de aptidões. Caso o jogador não atenda a esses requisitos, ele ainda pode fazer outras interações exceto a de batalha.