



CENTRO UNIVERSITÁRIO FEI - CAMPUS SÃO BERNARDO DO CAMPO
COORDENAÇÃO DE CURSO DE BACHARELADO EM CIÊNCIA DA
COMPUTAÇÃO
Banco de Dados (CCD110)

TRABALHO PRÁTICO - BANCO DE DADOS SIMPLIFICADO DO JOGO DE RPG
REALM OF THE MAD GOD

PEDRO SCHNEIDER - R.A. 24.124.072-0
GABRIEL SANTANA DIAS - R.A. 24.124.071-2

SÃO BERNARDO DO CAMPO - SP
2024

PEDRO SCHNEIDER - R.A. 24.124.072-0
GABRIEL SANTANA DIAS - R.A. 24.124.071-2

TRABALHO PRÁTICO - BANCO DE DADOS SIMPLIFICADO DO JOGO DE RPG
REALM OF THE MAD GOD

Projeto conceitual da base de dados do trabalho prático da disciplina Banco de Dados (CCD110), sob orientação do Prof. Rafael Luiz Testa.

SÃO BERNARDO DO CAMPO - SP
2024

RESUMO

O presente trabalho tem como objetivo recriar uma base de dados simplificada do *RPG Realm of The Mad God* (abreviado para *ROTMG*). O objetivo principal é facilitar o entendimento das operações fundamentais do jogo, bem como ilustrar de forma fácil o possível funcionamento real do banco de dados original e avaliar as relações entre usuário, monstros (NPCs, ou *Non-Playable Characters*), itens e *dungeons* ("masmorras", em tradução literal).

1 OBJETIVOS DO PROJETO

O conceito do banco de dados apresentado tem um objetivo muito simples: apresentar e exemplificar os fundamentos básicos de um *massive multiplayer online role-playing game*, ou, para abreviar, MMORPG. O modelo conta com entidades que representam usuários e seus personagens, itens, eventos, monstros e os biomas/masmorras onde podem ser encontrados, além das relações entre cada uma dessas entidades.

A modelagem leva em conta uma versão simplificada do funcionamento original do jogo e ignora alguns aspectos não tão importantes ou desnecessariamente complexos para o momento (como o funcionamento do *vault* (ou "cofre") e os *pets*). Somente com o modelo-entidade relacional apresentado, é possível entender em sua totalidade como, possivelmente, são armazenados os dados do jogo e como ele funciona em sua essência.

2 PRINCIPAIS FUNCIONALIDADES

Para o **usuário**:

- Criação de uma conta, usada para armazenar os personagens do jogador;
- Criação de personagens para uma conta, escolhendo a sua classe;
- Coleta e equipamento de itens, genéricos ou exclusivos de classe;
- Eliminação de monstros (relacionado à coleta de itens);

Para o **administrador do banco de dados**:

- Criação de itens;
- Criação de eventos e masmorras;
- Criação de monstros.

3 LÓGICA PARA MODELAGEM

Foram consideradas as ações que um usuário pode tomar com sua conta/personagens e a relação delas com a existência dos monstros.

- Um usuário possui uma **Conta** com um ID e e-mail únicos, um nome e uma senha;

*Uma **Conta** possui n **Personagens** únicos, porém um único Personagem só pertence a uma Conta.

- Um **Personagem** (nomeado como **CHAR** no modelo) possui seu ID único, seus atributos (uma tupla multivalorada contendo os seus valores de LIFE, MANA, ATTACK, DEFENSE, SPEED, DEXTERITY, WISDOM e VITALITY) e seu nível;

*Um **Personagem** pode ter 1 única **Classe**, porém uma classe pode pertencer a n Personagens.

*Um **Personagem** pode ter n **Itens** e um único Item pode pertencer a até n classes.

- Uma **Classe** possui um ID único (18 classes diferentes no total), seu nome, uma descrição e suas habilidades únicas;

*Uma **Classe** possui no mínimo 0 **Itens** e no máximo n , sendo alguns tipos de itens exclusivos para apenas algumas classes.

*Um **Item** pertence a no mínimo 1 **Classe** e no máximo n .

- Um **Item** possui um ID único, seu *tier* (nível que mede "o quão bom/raro" ele é), seu nome, um atributo que identifica se ele é trocável ou não e 5 subclasses: anel, poção, arma, armadura e especial, cada um com seu próprio efeito. As subclasses são **disjuntas**, ou seja, um anel não pode ser, ao mesmo tempo, uma poção.

*Na entidade **Item**, as subclasses **arma**, **armadura** e **especial** são específicas para cada classe (por exemplo: uma classe x usa uma arma x , portanto a arma y não pode ser utilizada por ela).

- Um **Monstro** possui um ID único, a quantidade de experiência que ele dá para os personagens que o eliminar, seu nome, seu nível e sua vida. Além disso, um monstro possui 3 subclasses **disjuntas**: monstro lacaio, que possui um atributo com o nome do bioma onde ele é encontrado; monstro de evento, relacionado à entidade **Evento**; e monstro chefe (*boss* no modelo), relacionado à entidade **Masmorra** (*Dungeon* no modelo);

*Um **Monstro** pode dar de 1 até n **Itens** ao ser eliminado e um Item pode pertencer a 1 ou n monstros.

- Um **Evento** possui seu ID único, seu nome e a duração.

*Um **Monstro de evento** pode pertencer a apenas 1 **Evento**, porém um Evento pode ter 1 ou n monstros.

- Uma **Masmorra** possui seu ID único, seu nome, sua dificuldade e os inimigos que podem ser encontrados dentro dela.

Um **Monstro chefe** pode pertencer a apenas 1 **Masmorra**, porém uma Masmorra pode ter n Monstros Chefes

Figura 1 – MER realizado considerando as funcionalidades e lógica apresentadas acima

