

Roadmap Técnico – RPG via Bot Discord

Este documento descreve a arquitetura, sistemas e boas práticas recomendadas para evolução de um RPG desenvolvido via bot no Discord, incluindo combate avançado, persistência e uso de SQLite.

1. Arquitetura Geral

Manter inicialmente arquivos JSON para persistência simples, centralizando leitura e escrita em um único módulo. Para médio prazo, migrar para SQLite visando estabilidade, concorrência e escalabilidade.

2. Sistema de Combate por Turnos

O combate deve operar em loop de turnos persistente, armazenando estado atual do jogador, inimigo, cooldowns e status negativos. Cada ação altera o estado até vitória, derrota ou fuga.

3. Habilidades Ativas e Cooldown

Cada habilidade possui custo, dano, efeito e tempo de recarga. Cooldowns são reduzidos a cada turno, impedindo uso contínuo e promovendo estratégia.

4. Status Negativos (Debuffs)

Implementar efeitos como sangramento, veneno, medo e atordoamento. Cada status possui duração e efeito aplicado automaticamente a cada turno.

5. Inimigos Elite e Chefes com Fases

Inimigos elite possuem atributos superiores e habilidades próprias. Chefes devem operar com fases baseadas em porcentagem de vida, alterando comportamento e habilidades.

6. Missões Persistentes

Missões devem possuir progresso salvo, objetivos claros e recompensa ao completar. Podem ser de caça, exploração ou narrativa dinâmica.

7. Uso do SQLite no Bot

O SQLite é recomendado para bots Discord por ser leve, rápido e não exigir servidor externo. Ele permite múltiplas leituras simultâneas com segurança.

Estrutura Básica de Tabelas:

- **players** (id, level, xp, hp, gold)
- **inventories** (player_id, item, quantity)
- **skills** (player_id, skill, cooldown)
- **status** (player_id, status, turns)
- **quests** (player_id, quest_id, progress, completed)

Recomenda-se criar um módulo *database.py* responsável por inicializar o banco, executar consultas e gerenciar conexões.

8. Boas Práticas

Validar dados ao carregar jogador, registrar logs de combate e evitar armazenar credenciais sensíveis no código ou repositório.

Conclusão

Com essa estrutura, o RPG evolui para um sistema robusto, escalável e pronto para longo prazo, permitindo expansão contínua de conteúdo e mecânicas.