

# Aura de Chumbo RPG — Evidências de Requisitos

## 1. Execução do Jogo

```
public static void main(String[] args) {  
    System.out.println("🎮 AURA DE CHUMBO - RPG SIMPLES 🎮");  
    // ...inicialização e menu...  
}
```

O jogo inicia pelo terminal, exibindo o menu principal.

## 2. Criação de Personagem (Herança, Enum, Composição)

```
public Personagem criarPersonagem(String nome, TipoPersonagem tipo) {  
    // ...  
    switch (tipo) {  
        case GUERREIRO:  
            personagem = new Guerreiro(nome);  
            break;  
        case MAGO:  
            personagem = new Mago(nome);  
            break;  
        case ARQUEIRO:  
            personagem = new Arqueiro(nome);  
            break;  
        default:  
            throw new IllegalArgumentException("Tipo inválido");  
    }  
    // ...  
}
```

Demonstra uso de enum, herança e composição (Jogador tem lista de Personagem).

## 3. Sistema de Batalha e Polimorfismo

```
switch (acao) {  
    case 1:  
        personagem.atacar(monstro);  
        break;  
    case 2:  
        personagem.usarHabilidadeEspecial(monstro);  
        break;  
    // ...  
}
```

O método `usarHabilidadeEspecial` é polimórfico, implementado de forma diferente em cada subclasse.

## 4. Validação e Mensagens de Erro

```
if (nome == null || nome.trim().isEmpty()) {
    System.out.println("❌ O nome não pode estar vazio!");
    return null;
}
for (Personagem p : personagens) {
    if (p.getNome().equalsIgnoreCase(nome)) {
        System.out.println("❌ Já existe um personagem com este nome!");
        return null;
    }
}
```

Valida nome vazio e duplicidade.

## 5. Progressão de XP e Level Up

```
public void ganharExperiencia(int xp) {
    experiencia += xp;
    while (experiencia >= experienciaParaProximoNivel()) {
        experiencia -= experienciaParaProximoNivel();
        nivel++;
        vidaMaxima += 10;
        ataque += 2;
        defesa += 1;
        vida = vidaMaxima; // cura completa ao upar
    }
}
```

Personagem ganha XP, sobe de nível e atributos aumentam.

## 6. Habilidades e Relacionamento N:N

```
private List<Habilidade> habilidades;

public void aprenderHabilidade(Habilidade habilidade) {
    if (!habilidades.contains(habilidade)) {
        habilidades.add(habilidade);
    }
}
```

Um personagem pode aprender várias habilidades e uma habilidade pode ser aprendida por vários personagens.

## 7. Finalização do Jogo

```
public void checarMorte() {
    boolean todosMortos = true;
    for (Personagem p : personagens) {
```

```
        if (p.estaVivo()) {
            todosMortos = false;
            break;
        }
    }
    if (todosMortos && !morto) {
        morto = true;
        System.out.println("\nSe fudeu");
    }
}
```

O jogo termina quando todos os personagens do jogador morrem.

## 8. Código-fonte e Documentação

```
// Exemplo de comentário minimalista
// Cria um personagem novo
public Personagem criarPersonagem(String nome, TipoPersonagem tipo) { ... }
```

O projeto está bem documentado e comentado.

## 10. Link do Repositório

<https://github.com/riessz/AuraDeChumbo>