Ethan\_Hueh

ToolsForRPG  
Panpalianos

Utilitário para Old Dragon

Sumário

[**1. Introdução** 2](#_Toc170244059)

[**2. Requisitos** 3](#_Toc170244060)

[**RF01 - Manter fichas** 3](#_Toc170244061)

[**RF02 - Leitura csv de fichas** 3](#_Toc170244062)

[**RF03 - Geração de estatísticas das fichas dos jogadores** 3](#_Toc170244063)

[**RF04 - Geração de tabela de preço** 3](#_Toc170244064)

[**RF05 – Gerenciar iniciativas de combate** 3](#_Toc170244065)

[**RNF01 – Linguagem de programação** 4](#_Toc170244066)

[**RNF02 – Acessibilidade** 4](#_Toc170244067)

[**3. Regras de Negócio** 5](#_Toc170244068)

[**Anexos** 6](#_Toc170244069)

[**Entidades** 6](#_Toc170244070)

[Anexo 9](#_Toc170244071)

# **1. Introdução**

Esse documento terá como objetivo documentar o software Tools For RPG – Panpalianos, um projeto pessoal que serve como utilitário para o RPG de Mesa Old Dragon.

# **2. Requisitos**

Esse tópico tem como fim reunir os requisitos funcionais, não funcionais e regras de negócio:

* RF: Requisito Funcional
* RNF: Requisito Não Funcional

# **RF01 - Manter fichas**

Temos a opção de cadastrar, consultar, atualizar e excluir fichas.

As opções de criação de ficha são as seguintes:

* Ficha genérica;
* Ficha de jogador;
* Ficha com atributos aleatórios;

# **RF02 - Leitura csv de fichas**

O sistema deverá ser capaz de ler um arquivo csv de fichas.

# **RF03 - Geração de estatísticas das fichas dos jogadores**

O sistema deverá ser capaz de gerar estatística das fichas dos jogadores, como:

* Jogador com maior atributo x;
* Jogador com menor atributo x;
* Média do atributo x na party;

# **RF04 - Geração de tabela de preço**

O sistema deverá ser capaz de gerar uma tabela de preço de um equipamento requisitado pelo usuário.

# **RF05 – Gerenciar iniciativas de combate**

O sistema deverá ser capaz de gerenciar as iniciativas num combate qualquer, levando em conta todas as criaturas presentes no combate, como jogadores e monstros.

# **RNF01 – Linguagem de programação**

O sistema será implementado em Java.

# **RNF02 – Acessibilidade**

O sistema será feito para que seja facilmente acessível pelo usuário, sendo leve e eficiente.

# **3. Regras de Negócio**

Esse tópico tem como fim descrever as regras de negócio do sistema, que nesse caso envolvem as regras do RPG Old Dragon.

# **Anexos**

# **Entidades**

1. **Ficha de criatura:**

Uma ficha genérica que representa uma criatura qualquer.

**Atributos:**

* Nome;
* Raça;
* Alinhamentos;
* Dados de vida;
* Movimento (em metros);
* Pontos de vida adicionais;
* Equipamentos;
* ATRIBUTOS, representado por:
* Força;
* Destreza;
* Constituição;
* Inteligência;
* Sabedoria;
* Carisma;
* SUBATRIBUTOS, representado por:
* Pontos de vida;
* Classe de armadura;
* Jogada de proteção;
* Base de ataque;
* Carga;

1. **Ficha de jogador (subclasse de ficha de criatura):**

Uma ficha genérica que representa uma criatura qualquer.

**Atributos adicionais:**

* Classe;
* Nível;

Interface gráfica do usuário

Descrição gerada automaticamente com confiança média

Interface gráfica do usuário

Descrição gerada automaticamente com confiança média

Notas:

\* Por favor. Você deve compilar esse projeto Java para uma versão mais

antiga (Java 8) para ser possível conseguir rodar aqui no PC.

Basicamente se você compilar para uma versão mais recente, como por

exemplo, é na JDK (que vai até o Java 21) a parada não vai abrir.

Enfim, acho que deu pra entender. Compile o projeto para Java 8.

# Anexo

**classDiagram**

    Ficha **\*--** Atributos

    Ficha **\*--** SubAtributos

    Ficha **\*--** Inventario

    FichaCriatura **--|>** Ficha

    FichaJogador **--|>** Ficha

    FichaJogador o**--** Classe

    Classe o**--** Especializacao

    Inventario o**--** Item

**class** Ficha {

        <<Abstract>>

        nome

        raca

        alinhamento

        dadosVida

        pontosVidaAtual

        movimento

        idiomas

        getPontosVida()

    }

**class** FichaJogador {

        nivel

    }

**class** FichaCriatura {

        pontosVidaAdicionais

    }

**class** Classe {

        <<enum>>

        nome

    }

**class** Especializacao {

        <<enum>>

        nome

    }

**class** Atributos {

        forca

        destreza

        constituicao

        inteligencia

        sabedoria

        carisma

    }

**class** SubAtributos {

        classeArmadura

        jogadaProtecao

        baseAtaque

    }

**class** Inventario {

        equipamentos

        calcularCarga()

    }

**class** Item {

        nome

        peso

    }