**Requisitos Funcionais**

1. **Cadastro de Personagem**

O sistema deve permitir criar um personagem informando:

* Nome
* Classe (ex: guerreiro, mago, arqueiro etc.)
* Atributos (Força, agilidade, vida). O usuário terá no máximo 10 pontos para serem distribuídos entre esses 3 atributos.

Após inserir as informações, o sistema deverá armazenar esse personagem em uma variável própria, com nome que foi inserido pelo usuário.

1. **Listagem de Personagens**

O sistema deve exibir uma lista com todos os personagens criados.

O sistema deve permitir selecionar um personagem para:

* Editar
* Excluir
* Exibir personagem

1. **Edição de Personagem**

O sistema deve permitir alterar os dados de um personagem existente.

Ao abrir a tela de edição, os dados atuais devem aparecer preenchidos.

O sistema deve salvar as alterações.

1. **Exclusão de Personagem**

O sistema deve permitir excluir um personagem da lista.

Deverá aparecer uma confirmação de exclusão antes de concluir a ação

1. **Organização de Equipes**

O sistema deve permitir selecionar até no máximo 3 e no mínimo 2 personagens para formar uma equipe.

O usuário deve poder visualizar os personagens escolhidos antes de confirmar a equipe.

O sistema deve salvar a equipe criada.

O usuário deverá conseguir selecionar e visualizar a equipe criada.

**Requisitos Não Funcionais**

1. **Interface**

A navegação deve ser intuitiva, com menus claros:

* Criar Personagem
* Listar Personagens
* Organizar Equipes

1. **Usabilidade**

A interface deve ser simples, permitindo fácil criação e edição dos personagens.

Confirmações de exclusão devem evitar exclusões acidentais.

1. **Manutenibilidade**

O código deve ser modular, com classes separadas para:

* Personagem
* Equipe
* GerenciarPersonagens
* GerenciarEquipes
* Interface