



## Sistema de Gerenciamento para o GDR

### 1. Introdução ao sistema

GDR (Guardiões das Relíquias) é uma campanha RPG com foco na trama e puzzle solving, neste RPG estaremos usando um sistema próprio para a campanha.

O sistema criado servirá para o gerenciamento geral do GDR, apesar de ainda não ter uma visão clara de todas as "features" do sistema, já tenho uma ideia das coisas que precisam ser incluídas.

### 2. Requisitos funcionais

**Requisitos do sistema acessíveis tanto aos jogadores quanto ao narrador:**

1. Gerenciar equipamentos e inventário;
2. Visualização de status com HP/STM/SNT;
3. Conter lista com a ficha de cada personagem;
4. Conter dados de rolagem;
5. Conter listagem de documentos encontrados;
6. Seção dedicada ao mapa;
7. *(Não confirmado) Ter acesso de usuário com senha;*



**Requisitos do sistema acessíveis apenas ao narrador:**

1. Criar ficha de jogadores com base em um modelo pré-definido;
2. Adicionar/Remover itens ou documentos de jogadores;
3. Adicionar/Remover jogadores;
4. *(Não confirmado) Remover acesso de login de jogadores;*



### **Requisitos ainda não confirmados ou necessitam de análise aprofundada**

1. Uma página que exibe detalhes do universo de GDR, como uma "Wikipédia" para os jogadores se imergir na história;
2. Gerenciador de "Buffs/Debuffs" disponível para o narrador;
3. Acessos com usuário e senhas para jogadores e para o narrador;
4. Gerenciador de turnos de cada jogador;
5. Exibir na página principal dos jogadores o turno de cada um, juntamente com quantos turnos se passaram;
6. Um possível sistema de crafting;
7. (Precisa analisar o tamanho do escopo) Um sistema de mensagens, que os jogadores e o narrador poderão enviar para os outros;

### **3. Requisitos não funcionais**

1. O sistema deve ser acessível através de dispositivos móveis e computadores através de um servidor local via flask;
2. UI a princípio precisa seguir um design pixel-art/hacker-like;
3. O sistema deve ser capaz de lidar com até 10 usuários simultaneamente sem perda significativa de desempenho, considerando o escopo limitado do servidor local;
4. O sistema deve ser executado em navegadores modernos (Google Chrome, Firefox e Edge) sem necessidade de plugins adicionais;
5. A interface deve ser responsiva para suportar telas de diferentes tamanhos (smartphones, tablets e desktops);
6. O sistema deve ser modular o suficiente para facilitar a adição de novos recursos no futuro, como o sistema de mensagens ou crafting, sem necessidade de reescrever partes significativas do código;
7. O design deve ser intuitivo, permitindo que jogadores e narradores acessem as funcionalidades principais sem um treinamento prévio detalhado;
8. O código deve ser documentado e seguir boas práticas de programação para facilitar futuras correções e atualizações;
9. O sistema deve utilizar Flask para o backend e respeitar o padrão de rotas RESTful para as APIs;



10.0 sistema deve ser capaz de lidar com erros comuns, como entradas inválidas ou desconexão de jogadores, sem comprometer a integridade dos dados;

11.0s dados gerados pelo sistema (fichas, inventários, logs) devem ser exportáveis em formatos comuns, como JSON ou CSV, para backup ou edição offline;



GDR