PROJETO PARA AVALIAÇÃO TÉCNICA

Desenvolvedor: Victor Aparecido Ribeiro

Bauru/SP 2019

Taverna de personagens

Desenvolvedor: Victor Aparecido Ribeiro

Relatório de documentação sobre o projeto desenvolvido.

Bauru/SP 2019

1.	TECNOLOGIAS UTILIZADAS	4
2.	LOGIN	4
3.	PERFIL ADMINISTRADOR	5
4.	PERFIL DO USUÁRIO	6
5.	CADASTRO DE CLASSES	7
6.	CADASTRO DE ESPECIALIDADES	8
7.	CADASTRO DE PERSONAGENS	9
8.	API	. 11
9.	INFORMAÇÕES ADICIONAIS	. 15

1. TECNOLOGIAS UTILIZADAS

O projeto foi desenvolvido em **Java**, utilizando os frameworks para desenvolvimento web **JSF** + **PrimeFaces** em conjunto com o template visual **Ecuador** do próprio **PrimeFaces**. O **Hibernate** foi utilizado como ORM para mapeamento e manuseio do banco de dados, bem como o **JPA** para a persistência de dados.

A api foi desenvolvida utilizando o padrão REST e documentada com a ferramenta **Swagger.** A base de dados escolhida foi o banco relacional **PostgreSQL.**

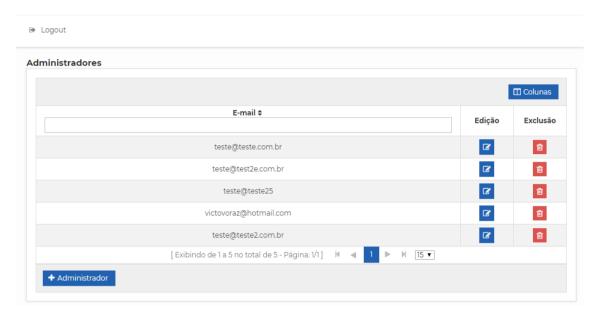
2. LOGIN

Logando com o **usuário: 0 e senha: verbatim**@@*, você terá acesso ao menu admin e poderá cadastrar novos usuários. Já para logar com os usuários comuns, você deve utilizar o e-mail e senha cadastrado no perfil do admin.

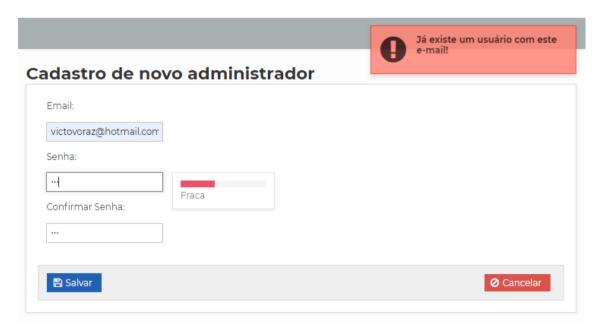


3. PERFIL ADMINISTRADOR

No perfil do administrador uma listagem dos usuários cadastrados no sistema será exibida. O perfil possui todas as funções do CRUD(cadastro, edição e exclusão), bem como uma pesquisa por e-mail dos usuários.

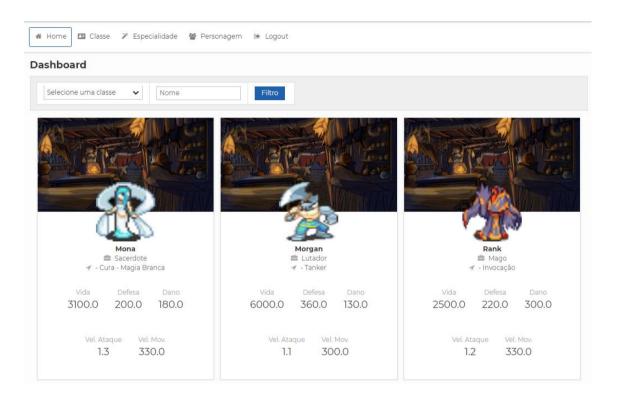


No cadastro de novo usuário administrador é solicitado o e-mail, senha (e confirmação de senha). Uma validação é aplicada para não permitir que dois usuários possuam o mesmo e-mail.



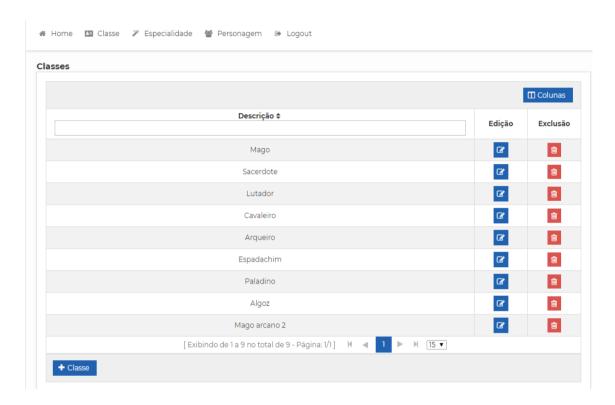
4. PERFIL DO USUÁRIO

No perfil do usuário, as opções de cadastro de **Classe, Especialidade e Personagem** estão disponíveis. Na **Dashboard**, os personagens cadastrados serão listados exibindo seus status. Um filtro de pesquisa por **nome e classe** também está disponível.



5. CADASTRO DE CLASSES

Uma listagem exibindo todas as classes será exibida, as funções do CRUD(cadastro, edição e exclusão) estarão disponíveis, bem como uma pesquisa pela descrição das classes.

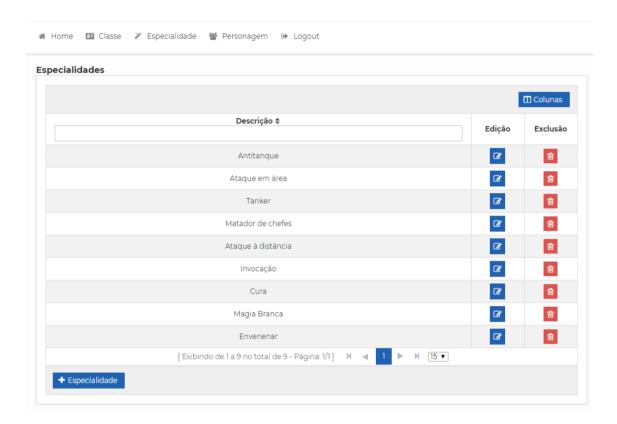


O cadastro solicita ao usuário apenas a descrição da classe e faz uma validação impedindo que a descrição seja replicada.



6. CADASTRO DE ESPECIALIDADES

Uma listagem exibindo todas as especialidades será exibida, as funções do CRUD(cadastro, edição e exclusão) estarão disponíveis, bem como uma pesquisa pela descrição das especialidades.

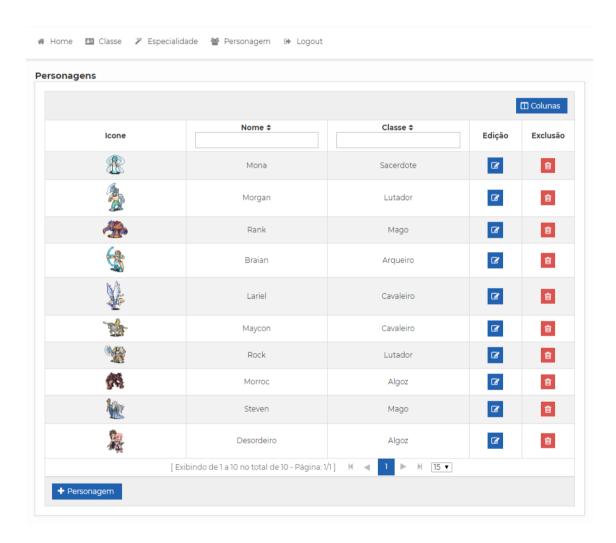


O cadastro solicita ao usuário apenas a descrição da especialidade e faz uma validação impedindo que a descrição seja replicada.



7. CADASTRO DE PERSONAGENS

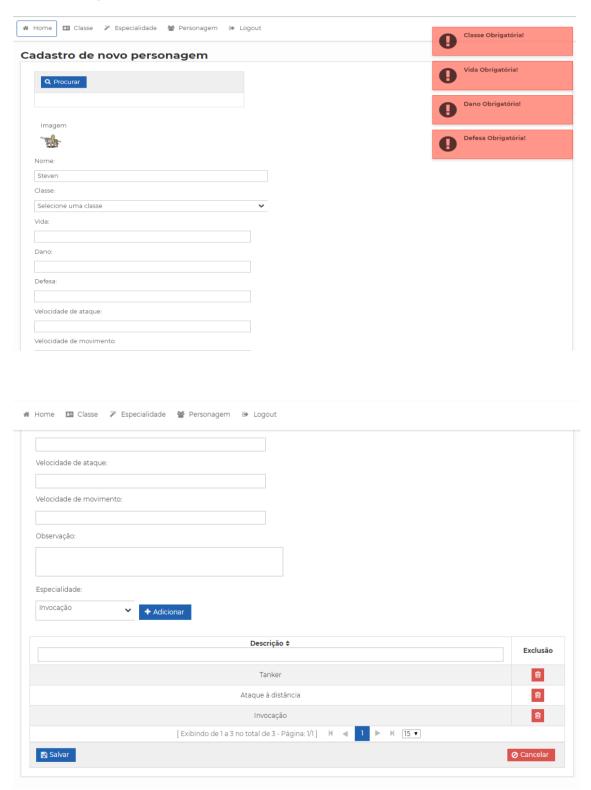
Uma listagem exibindo todas os personagens será exibida, as funções do CRUD(cadastro, edição e exclusão) estarão disponíveis, bem como uma pesquisa pelo nome e classe dos personagens.



No cadastro de personagem os campos solicitados serão: Imagem, Nome, Classe, Vida, Defesa, Velocidade de ataque, Velocidade de movimento, Observação e

Especialidade (Podendo ser 1 ou N). Porém só os campos Nome, Classe, Vida, Dano e Defesa são obrigatórios.

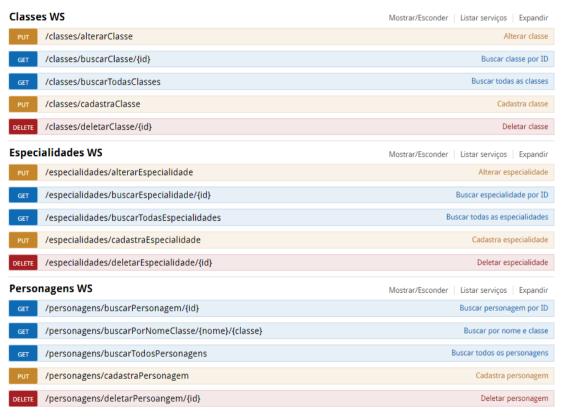
O sistema também faz uma validação para impedir que o existam personagens com nomes iguais.



8. **API**

Acessando o caminho taverna/doc/index.html você terá acesso a API do sistema.

Documentação API



[BASE URL: /taverna/api , Versão API: 1.5.3]

9. INFORMAÇÕES ADICIONAIS

Dentro da pasta do projeto, uma pasta chamada **banco** armazenará o script de criação do banco de dados, bem como o arquivo de backup do banco.

O projeto também foi disponibilizado em um ambiente de desenvolvimento que hospedo na OVH. Creio que para visualizar o sistema será mais fácil se acessarem por lá.

Link do sistema: http://51.79.28.213/taverna

Link da api: http://51.79.28.213/taverna/doc/index.html