1. StoryTelling

O cliente HROADS deseja começar a construir o seu próprio jogo de RPG online. Para isto, ele definiu que cada personagem do jogo, possuirá uma classe e que cada classe do jogo irá possuir uma ou mais habilidades, e esta habilidade pertence somente a um tipo de habilidade.

Por exemplo:

O personagem **DeuBug** é da classe Bárbaro que por sua vez possui uma habilidade de ataque chamada Lança Mortal e uma habilidade de defesa chamada Escudo Supremo.

DeuBug	-> classe Bárbaro	-> Lança Mortal	-> ataque
		-> Escudo Supremo	-> defesa

O personagem **BitBug** é da classe Monge que por sua vez possui uma habilidade de cura chamada Recuperar Vida e uma habilidade de defesa chamada Escudo Supremo.

BitBug	-> classe Monge	-> Recuperar Vida	-> cura
		-> Escudo Supremo	-> defesa

Cada personagem deve pertencer exclusivamente a uma única classe. Uma classe pode ter uma ou mais habilidades. E uma habilidade deve pertencer a exclusivamente um único tipo de habilidade.

Para que o jogo tenha início, HROADS disponibilizou um conteúdo com algumas informações sobre como ele quer que o jogo seja construído.

Classes

- Bárbaro (Lança Mortal, Escudo Supremo)
- Cruzado (Escudo Supremo)
- Caçadora de Demônios (Lança Mortal)
- Monge (Recuperar Vida, Escudo Supremo)
- Necromante (começa sem habilidades)
- Feiticeiro (Recuperar Vida)
- Arcanista (começa sem habilidades)

Habilidades

- Lança Mortal (tipo de habilidade: ataque)
- Escudo Supremo (tipo de habilidade: defesa)
- Recuperar Vida (tipo de habilidade: cura)

Tipos de Habilidades

- Ataque
- Defesa
- Cura
- Magia

Personagens

Nome Personagem: DeuBug

Classe: Bárbaro

Capacidade Máxima Vida: 100

Capacidade Máxima Mana: 80

Data de Atualização: Data Atual

Data de Criação: 18/01/2019

Nome Personagem: BitBug

Classe: Monge

Capacidade Máxima Vida: 70

Capacidade Máxima Mana: 100

Data de Atualização: Data Atual

Data de Criação: 17/03/2016

Nome Personagem: Fer8

Classe: Arcanista

Capacidade Máxima Vida: 75

Capacidade Máxima Mana: 60

Data de Atualização: Data Atual

Data de Criação: 18/03/2018

Roteiro API

O cliente **HROADS** aprovou o desenvolvimento do banco de dados e gostou do seu trabalho, por isso lhe contratou para desenvolver também a solução back-end utilizando a interface API.

Para isso, você deverá:

Criar um novo projeto do tipo WebAPI com o seguinte nome: **senai.hroads.webApi**.

Para a construção, é necessário utilizar o **Entity Framework Core**.

A abordagem será de sua escolha: **Database First** ou **Code First**.

Sua API deverá ter as seguintes funcionalidades:

- CRUD completo de todas as entidades (Tipos de usuário, Usuários, Classes, Tipos de Habilidade, Habilidades e Personagens);
- Para as listagens, os dados das entidades relacionadas também deverão ser mostrados. Por exemplo: ao listar uma **Habilidade**, também deverá ser mostrado o título do **Tipo de Habilidade** ao qual esta se refere;
- Somente o usuário do tipo **ADMINISTRADOR** poderá realizar os cadastros de todas as entidades, exceto a de **Personagens**;
- Somente o usuário do tipo **JOGADOR** poderá cadastrar um novo personagem (não é necessário atrelar um personagem a um jogador).
- O usuário do tipo **JOGADOR** ou **ADMINISTRADOR** poderá visualizar a lista com todos os personagens;
- A visualização das **Classes**, **Tipos de Habilidade** e **Habilidades** do jogo é **pública**.
- Sua autenticação deverá ser feita utilizando **JWT** (não esquecer de criar um endpoint para gerar o token);
- Utilizar o **Swagger** para a documentação de sua API;
- Na resposta da API em formato JSON, os valores nulos deverão ser ignorados.

Dica

Crie um **LoginController** para gerar o token e assim separar esta funcionalidade dos endpoints de **Usuários**.

Desafios extras da API

- Quando um jogador criar um personagem, atrelar este personagem a este jogador;
- Listar todos os personagens de um determinado jogador;
- Criar um endpoint para que somente o administrador tenha acesso e consiga visualizar a lista de todos os personagens com os dados de seus respectivos jogadores;
- Ao listar todos os jogadores, **não** mostrar suas senhas;
- Listar todos os personagens cadastrados ordenados pela Classe em ordem alfabética crescente (A – Z).

O que deverá ser entregue?

Deverá ser compartilhado o link do GitHub com a pasta do desafio criada dentro do repositório da sprint 2 – back-end, contendo a seguinte estrutura de pastas:

BD

- Os arquivos com os scripts DDL, DML e DQL (no caso de Database First);
- Os arquivos com as modelagens Conceitual, Lógico e Físico;
- O Diagrama de banco de dados exportado pelo SSMS.

Back-end

Solução do projeto (não esqueça que o arquivo .sln **não** armazena os demais arquivos do seu projeto, é preciso entregar toda a estrutura criada).

Postman

Exportar a coleção de requisições criada com a estrutura de pastas:

- Tipos de Usuário;
- Usuários;
- Classes;
- Tipos de Habilidade;
- Habilidades;
- Personagens.

Planejamento

Faça a entrega também do link do quadro do Trello com todo o planejamento das tarefas desenvolvidas.