

Jogo da Velha

Para o desenvolvimento da api foi escolhido o micro-framework Flask, apesar de possuir um núcleo simples, é extensível, possibilita a construção de apis simples à apis com alto nível de complexidade e escaláveis. Um requisito fundamental e desejável para o entendimento do serviço é a documentação, o flask possui extensões facilmente configuráveis como swagger. Será construído uma documentação interativa em um arquivo específico baseado na especificação openApi.

Organização do código

Existem muitos modelos e variação de padrões diferentes usando para a construção de apis usando o flask. Para fins de organização e separação de responsabilidades na api, será usando da seguinte forma:

- api**
 - specs**
 - src**
 - controllers**
 - models**
 - helpers**
 - service**
 - tests**
 - static**
 - auth**
 - app**
 - config**

Foi definido um arquivo para construção do aplicativo e um para configuração com objetos de configuração para cada ambiente de execução. Além disso, um repositório para controladores, models responsáveis pela definição de payloads e validação, os módulos do models também são os responsáveis por salvar as persistências dos dados. Um módulos para possíveis serviços, auth para possível autenticação. A static é usada para disponibilizar os arquivos públicos, nesse caso foi utilizado para disponibilizar a documentação openApi do swagger. Alguns testes foram incluídos no projeto para garantir que novas mudanças adicionadas vão sempre estar de acordo com funcionamento do projeto. Na pasta raiz da api foi configurado arquivos de execução e definições para publicação do aplicativo.

O jogo

A variável **round_game** é que vai contabilizar o número de rodadas. É sempre iniciada com . Essa variável vai ser incrementada de um em um, a cada rodada. **round_player** é usado para a vez do jogador, representado por **X** ou **O**.

A cada jogada é alternado para outro jogador. Os jogadores são representados por **X** e **O**.

A classe **Game** representa todas funcionalidades para controlar o jogo. Se alguma linha, coluna ou diagonal for completada é retornado o símbolo do jogador que ganhou a partida, caso termine o número máximo de jogadas e nenhuma condição for satisfeita, é retornado o valor **Draw** representando empate.

Para que a matriz seja compatível com o tabuleiro, foi usando rotação de matriz. Ao final de cada movimento na partida, a matriz é rotacionada 90 graus no sentido anti-horário, para que os movimentos sejam inseridos na posição correta, antes de cada movimento no tabuleiro, a matriz é rotaciona 3 vezes, ou seja, rotacionada 270 graus no sentido anti-horário, desse modo a matriz voltará para o formato padrão antes de cada movimentação no tabuleiro.