# Critérios de teste utilizados

Para testar o projeto, um arquivo de teste foi utilizado, rodado usando pytest.

O arquivo test\_fis.py contém os testes para os 4 módulos, e foi criado usando uma liga de NBA Fantasy como exemplo.

Os critétios utilziados para os testes tiveram o intuito de conferir se cada módulo rodou corretamente, ou seja, o setup, a extração dos dados, a organização dos dados e a visualização dos dados foram executados corretamente com os dados da liga de exemplo.

## Módulo SetUp:

O SetUp é inicializado a partir de informações para o tipo de esporte, o ano e o identificador de uma liga específica.

Para verificar se o objeto da classe SetUp foi inicializado corretamente foram usados 3 testes:

* Test\_setup\_league\_name
* Test\_setup\_number\_teams
* Test\_setup\_league\_type

**test\_setup\_league\_name**

Verifica se o setup foi inicializado com o nome da liga correspondente ao código passado.

**test\_setup\_number\_teams**

Verifica se o setup foi inicializado com o número de times correspondente ao código passado.

**test\_setup\_league\_type**

Verifica se o setup foi inicializado com o tipo de liga correspondente ao código passado.

## Módulo Extract\_Data

O Extract\_Data é inicializado a partir de um objeto da classe SetUp, e extrai os dados da liga correspondente.

Para verificar se o objeto de Extract\_Data foi inicializado corretamente, 2 testes foram realizados:

* test\_extraction\_player\_count
* test\_extraction\_week\_1

**test\_extraction\_player\_count**

Verifica se o objeto criado contém o atributo “players\_per\_team” com o valor correspondente à configuração da liga usada para teste.

**test\_extraction\_week\_1**

Carrega o arquivo json com os dados extraídos pela classe, usando o caminho para o diretório contido como atributo do objeto, e verifica se os dados extraídos pelo objeto estão corretos. Essa checagem é feita ao observar um elemento específico do conjunto de dados extraídos, para ver se o seu valor bate com o valor correspondente para a liga.

## Módulo Data\_Organizer

O Data\_Organizer é inicializado a partir do mesmo objeto da classe SetUp, lê os dados extraídos para a liga correspondente e organiza os mesmos em um objeto da calsse DataFrame da biblioteca pandas.

Para verificar se o objeto de Data\_Organizer (Teams\_Weekly\_Data\_Organizer) foi inicializado corretamente, 3 testes foram realizados:

* test\_organizer\_current\_week
* test\_organizer\_data\_output
* test\_organizer\_week\_1

**test\_organizer\_current\_week**

Verifica se o objeto criado contém o atributo “current\_week” com o valor correspondente ao objeto da classe SetUp usado para inicializar a classe.

**test\_organizer\_data\_output**

Verifica se os atributos contendo os outputs da classe são do tipo correto, um objeto da classe DataFrame, da biblioteca pandas.

**test\_organizer\_week\_1**

Verifica se o DataFrame criado para organizar os dados está com os valores corretos. Essa checagem é feita ao observar um elemento específico do conjunto de dados extraídos, para ver se o seu valor bate com o valor correspondente para a liga.

## Módulo Data\_Visualization

O objeto de Data\_Visualization é inicializado a partir do objeto da classe Data\_Organzier com os dados estruturados, usa os dados vindos de um DataFrame da biblioteca pandas e cria visualizações a partir deles.

Para verificar se o objeto de Data\_Visualization, duas etapas foram feitas, uma para a sub-classe Visualize\_Teams\_Week\_Data e outro para a sub-classe Visualize\_Teams\_Season\_Data.

Para verificar se o objeto da classe Visualize\_Teams\_Week\_Data foi inicializado corretamente, 4 testes foram realizados, todos ao gerar visualizações com os dados da primeira semana:

* test\_visualizer\_week\_1\_table
* test\_visualizer\_week\_1\_table\_correctedness
* test\_visualizer\_week\_1\_score
* test\_visualizer\_week\_1\_score\_correctedness

**test\_visualizer\_week\_1\_table**

Verifica se se o atributo da classe contendo o output week\_table é do tipo correto, um objeto da classe Styler, da biblioteca pandas.

**test\_visualizer\_week\_1\_table\_correctedness**

Verifica se o output week\_table foi criado com os valores corretos. Essa checagem é feita ao observar o mesmo elemento específico utilizado anteriormente do conjunto de dados extraídos, para ver se o seu valor bate com o valor correspondente para a liga.

**test\_visualizer\_week\_1\_score**

Verifica se se o atributo da classe contendo o output week\_score é do tipo correto, um objeto da classe Styler, da biblioteca pandas.

**test\_visualizer\_week\_1\_score\_correctedness**

Verifica se o output week\_score foi criado com os valores corretos. Essa checagem é feita ao observar o mesmo elemento específico utilizado anteriormente do conjunto de dados extraídos, para ver se o seu valor bate com o valor correspondente para a liga.

Para verificar se o objeto da classe Visualize\_Teams\_Season\_Data foi inicializado corretamente, 2 testes foram realizados:

* test\_visualizer\_season\_table
* test\_visualizer\_season\_score

**test\_visualizer\_season\_table**

Verifica se se o atributo da classe contendo o output season\_table é do tipo correto, um objeto da classe Styler, da biblioteca pandas.

**test\_visualizer\_season\_score**

Verifica se se o atributo da classe contendo o output season\_table é do tipo correto, um objeto da classe AxesSubplot, da biblioteca matplotlib.