**Logotipo, nome da empresa

Descrição gerada automaticamente**Uma imagem contendo desenho, placar

Descrição gerada automaticamente

## Exercício Pratico Final – Instruções

ITA - TDD – Desenvolvimento de Software Guiado por Testes (Coursera)

Aluno: Renato Sampaio Albuquerque

Nesta tarefa será utilizado o TDD para desenvolver um componente de gamificação. Esse componente deve armazenar diferentes tipos de pontos que o usuário pode receber. Por exemplo, uma aplicação pode possuir pontos do tipo "moeda" e "estrela", enquanto outra pode possuir pontos do tipo "topico", "comentario" e "curtida". Esse componente deve possuir uma classe principal chamada Placar, onde deve ficar a lógica, e uma classe chamada Armazenamento, que deve ser responsável por guardar e recuperar as informações de um arquivo.

A classe **Armazenamento** deve ser capaz de realizar as seguintes operações:

* Armazenar que um usuário recebeu uma quantidade de um tipo de ponto. Por exemplo: o usuário "guerra" recebeu "10" pontos do tipo "estrela"
* Recuperar quantos pontos de um tipo tem um usuário. Por exemplo: retornar quantos pontos do tipo "estrela" tem o usuário "guerra"
* Retornar todos os usuários que já receberam algum tipo de ponto.
* Retornar todos os tipos de ponto que já foram registrados para algum usuário.

*Observação: os dados devem ser armazenados em um arquivo e como serão armazenados fica a critério do aprendiz. A seção "Formas de implementar o armazenamento em arquivo" dá algumas sugestões.*

A classe **Placar** é composta por uma instância de Armazenamento, a quem delega a recuperação e o armazenamento das informações. A classe Placar deve ter métodos que executam as seguintes operações:

* Registrar um tipo de ponto para um usuário. Por exemplo: o usuário "guerra" recebeu "10" pontos do tipo "estrela"
* Retornar todos os pontos de um usuário. Por exemplo: ao pedir os pontos do usuário "guerra" ele me retornaria que possui "20" pontos do tipo "moeda" e "25" pontos do tipo "estrela". Um tipo de ponto que o usuário não possuir, não deve ser retornado com valor "0". Por exemplo: se o usuário "guerra" não possui pontos do tipo "energia", esses não devem ser incluídos na resposta.
* Retornar ranking de um tipo de ponto, com a lista de usuário que possuem aquele ponto ordenados do que possui mais para o que possui menos. Por exemplo: ao pedir o ranking de "estrela", seria retornado "guerra" com "25", "fernandes" com "19" e "rodrigo" com "17". Um usuário que não possui pontos daquele tipo não seria incluído no ranking. Por exemplo, o usuário "toco" sem pontos do tipo "estrela" não seria incluído.

Os testes da classe Armazenamento devem ser feitos utilizando arquivos e os testes da classe Placar devem ser feitos criando um mock object para a instância de Armazenamento. Por fim, devem ser criados alguns testes de integração incluindo as duas classes.

A criação de outras classes e a assinatura dos métodos fica a critério do aprendiz, desde que cumpra os requisitos solicitados. Com a exceção de construtores e métodos de acesso, nenhum outro método público pode ser adicionado nas classes Armazenamento e Placar.

Não esqueça de utilizar o TDD para o desenvolvimento e de refatorar sempre o código para manter a sua qualidade!

**Review criteria**

**menos**

Você será avaliado com base no seguinte:

* Cumprimento dos requisitos de implementação pedidos no enunciado
* Organização do código implementado
* Uso correto de Mock Objects
* Divisão correta de responsabilidades entre as classes

**Formas de implementar o armazenamento em arquivo**

**menos**

É deixado livre a forma como os dados de pontuação do usuário serão armazenado em um arquivo, desde que os requisitos sejam cumpridos. É importante que outras classes não dependam de forma alguma de como é feita essa armazenagem no arquivo.

Uma abordagem para armazenar os dados seria fazer isso de uma forma incremental. Sempre só adicionando dados no arquivo. Nesse caso, para saber a pontuação de um usuário, seria necessário percorrer todo o arquivo procurando por todos os dados a respeito dele.

Outra abordagem seria ter um registro para cada usuário dentro do arquivo e modificar esse registro à medida que novos dados forem chegando.

Independente da abordagem de armazenar no arquivo, você pode também guardar um cache dos dados em memória. O único requisito nesse caso é que se a aplicação cair, deve-se recuperar todos os dados armazenados.

A pesquisa faz parte da rotina de qualquer profissional de computação, sendo assim, é considerado parte desse exercício a pesquisa das classes que podem ser utilizadas para a interação com arquivos. Lembre-se de ir seguindo o TDD e fazendo a implementação passo a passo. Você verá que assim é bem mais fácil de utilizar uma classe que você ainda não conhece.