

### AVALIAÇÃO III (Prática)

Você é o desenvolvedor líder de um sistema de software para o gerenciamento de competições de Corrida de Orientação. Na terceira fase do processo desenvolvimento o enfoque principal será no módulo de registro e apuração de percurso.

As principais classes envolvidas nesse módulo são:

- **BoletimProva:** Contém os dados de desempenho de um atleta em um percurso, incluindo o código CBO de um atleta e sua **ListaPassagens**, que consiste de uma lista de tuplas (lista de **TempoPassagens**), onde cada tupla registra o número de um prisma e o tempo em que o atleta o alcançou (em relação a sua largada), o tempo de largada previsto para o atleta e o tempo de .

Para garantir a correção do módulo de registro de percurso, o sistema deve cumprir os requisitos apresentados a seguir.

#### Questão I

O **BoletimProva** deve ter um comportamento dependente do momento da prova do atleta. A prova do atleta possui quatro fases: o **Pré Prova**, antes do atleta se apresentar pra largada; o **Momento da Largada**, quando o atleta se apresentou para a largada, mas ainda não registrou a Largada; a fase **Pista**, que acontece após o registro da largada e antes do registro da chegada; e por fim o **Pós Prova**, após o atleta ter registrado a chegada.

Na fase de **Pré Prova**, o registro de largada, e consequentemente de eventual tempo de atraso, o registro de chegada, e da passagem nos prismas, não é permitido.

Na fase de **Momento da Largada**, o atleta pode eventualmente registrar atraso (podendo consequentemente calcular o tempo de atraso como a subtração do tempo de partida efetivo menos o tempo de partida previsto), mas não pode registrar a passagens, nem a chegada.

Na fase de **Pista**, o boletim pode registrar o atraso, as passagens no prisma e a chegada.

No **Pós Prova**, após a chegada, o boletim não pode mais registrar passagens no prisma, mas ainda deve permitir o registro de atraso.

A classe **BoletimProva** foi disponibilizada com essas funcionalidades, mas mostra-se distante do desejável para garantir a manutenibilidade, o open/closed e a extensibilidade do sistema. É tarefa de sua equipe identificar e aplicar os padrões de projeto adequados refatorando o código da classe.

#### Questão II

A classe **Apurador** é responsável por receber um **BoletimProva**, proceder a apuração e determinar o tempo de prova do atleta. A classe foi construída de forma a considerar as seguintes regras: i) definição do tempo de prova base como o tempo de registro de chegada; ii) verificação do tempo máximo de prova; iii) verificação de registro do prisma na ordem correta, definida na **ListaPassagem**; iv) verificação se todos os prismas foram registrados; v) e penalização de atraso na partida com adição do tempo de atraso, ao tempo final.

Mais uma vez, a abordagem monolítica, está dificultando a adaptação do apurador de acordo com as características da prova, categoria, etc. Por exemplo, para algumas provas, pode se querer retirar a exigência de ordem dos prismas, mantendo-se a necessidade de verificação de registros em todos.

Para outras, pode-se retirar o tempo máximo e não penalizar atrasos em relação à lista de partida. Ou seja, uma vez que a regra esteja presente na versão do apurador para uma dada prova, ainda assim a sua aplicação de uma regra estará condicionada a verificação de sua aplicabilidade, através dos dados constantes do boletim. Só então, a regra verifica se é o caso de indicar que a prova não foi concluída (como é o caso do não registro de chegada), ou de adição de tempo (como é o caso do atraso na largada).

Mais uma vez, é tarefa de sua equipe identificar e aplicar os padrões de projeto adequados refatorando o código da classe, de forma a permitir que o apurador seja configurável, garantindo, inclusive, que novas regras possam ser adicionadas sem impacto para as existentes.

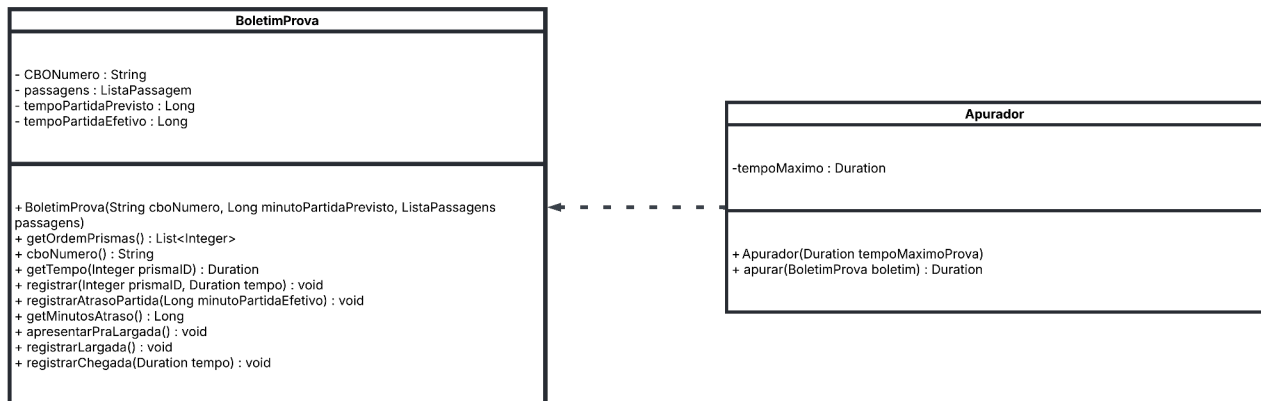


Figura 1: Detalhamento da Classe BoletimProva e Apurador

## Normas de Entrega

1. O trabalho deverá ser feito em equipe de até três pessoas e preferencialmente de três pessoas.
2. A entrega deve ser feita mediante a disponibilização do projeto em ambiente git (github) com acesso compartilhado para o usuário pooinf008
3. A data do último commit NÃO pode ser posterior ao dia 04/08/2025 às 23:59
4. O link do projeto no github deve ser remetido para o endereço eletrônico [poo.inf008@gmail.com](mailto:poo.inf008@gmail.com)
5. Este mail deverá conter no assunto (subject): [inf0112025.1aval3] <participante1> - <participante2> - <participante3>
6. O assunto será usado para filtragem automática, portanto atentem-se ao formato indicado.
7. Os padrões de projeto aplicados devem ser detalhados no readme.md, incluindo:
  - O nome do padrão de projeto.
  - Uma breve justificativa para sua escolha.
  - A identificação clara de cada classe do projeto e qual papel (participante) ela assume no padrão (ex: Memento, Originator, Caretaker)