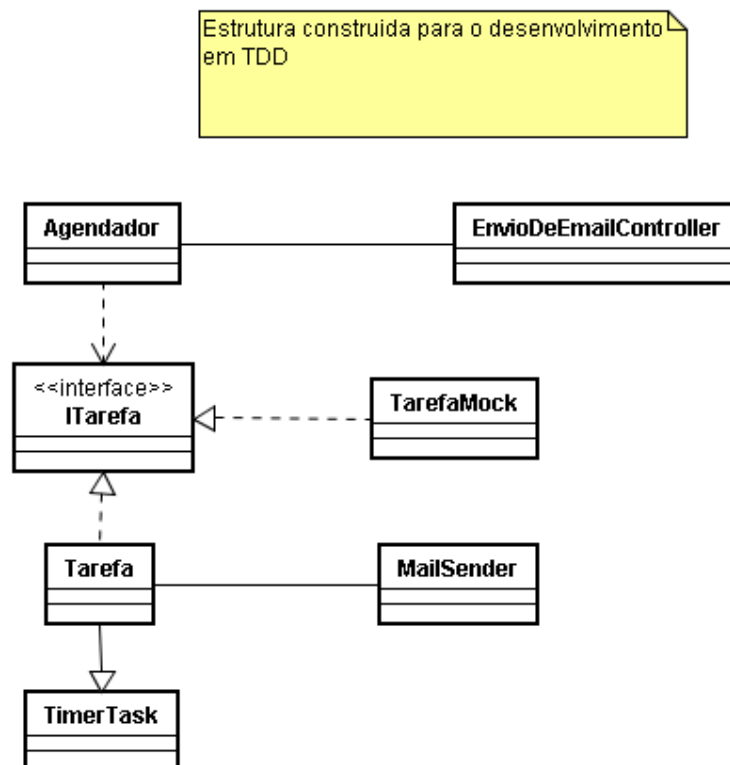


Sobre as classes desenvolvidas dirigidas por teste:

- Foram desenvolvidas três classes seguindo a disciplina do TDD:
 - **Tarefa:** Classe que implementa o TimerTask, classe de agendamento de eventos. Esta classe fará o agendamento do envio de e-mails.
 - **Agendador:** Classe que implementa todas as formas de envio de e-mail necessárias ao CRM. Este possui os métodos de envio de email em um dia marcado, em uma faixa de dias e para o envio de e-mail exceto nos fins de semana.
 - **EnvioDeEmailController:** Classe que prepara os dados recebidos do CRM para as classes de agendamento e envio de e-mails. Esta classe deve fazer todas as validações necessárias para que as requisições de envio de e-mails sejam feitas de forma correta. Ele deve verificar, por exemplo, os dias do envio, se o e-mail de envio está correto, se o assunto e o corpo da mensagem de e-mail foi escrito, dentre outros.



estrutura desenvolvida dirigida por testes

Como pode visto no diagrama acima, a classe tarefa utiliza-se das classes TimerTask, já citada acima e da classe de envio de E-mails MailSender. Esta classe já havia sido implementada anteriormente, por isto, não entrará no desenvolvimento.

Como a classe Tarefa possui algumas dependências (como por exemplo, estar conectada a internet, para que esta consiga completar o envio de e-mail), foi criada também a classe

TarefaMock, para que estas dependências não interfiram nos testes executados nas demais classes.

Sobre os relatórios dos testes:

Para o desenvolvimento das classes, foram criados os arquivos de testes, e três passos foram destacados:

No primeiro passo, existia uma classe de teste para a classe a ser desenvolvida, mas não existia a classe a ser testada.

No segundo, existiam as classes de testes e a classe a ser implementada, porém, esta classe possuía somente o código necessário para que a classe funcionasse, não executando sua função e, por consequência, nem todos os casos de testes foram aprovados.

No terceiro passo, foi feita a implementação completa do caso de uso. Isto foi feito gradualmente, de forma que todos os casos de testes produzidos fossem aprovados com sucesso.

Para cada uma das classes, foi feito um conjunto de arquivos possuindo relatórios gerados para cada um dos passos. Estes relatórios foram obtidos através do eclEmma, ferramenta de testes de cobertura da IDE eclipse. O objetivo destes relatórios de cobertura de código foi de apresentar a evolução do desenvolvimento das classes, até o ponto final, onde todos os testes passaram.