Enviar até Nov 23, 5:59 AM -02

**Envie sua tarefa em breve**

Mesmo que o prazo da tarefa seja até Nov 23, 5:59 AM -02, tente enviá-la com um ou dois dias de antecedência, se possível. Assim, você tem uma chance maior que conseguir a avaliação necessária do seu colega a tempo.

1. [**Instruções**](https://www.coursera.org/learn/tdd-desenvolvimento-de-software-guiado-por-testes/peer/XEGmZ/software-de-caixa-eletronico)
2. [**Meus envios**](https://www.coursera.org/learn/tdd-desenvolvimento-de-software-guiado-por-testes/peer/XEGmZ/software-de-caixa-eletronico/submit)
3. [**Discussões**](https://www.coursera.org/learn/tdd-desenvolvimento-de-software-guiado-por-testes/peer/XEGmZ/software-de-caixa-eletronico/discussions)

Criar, utilizando TDD, uma classe chamada CaixaEletronico, juntamente com a classe ContaCorrente, que possuem os requisitos abaixo:

* A classe CaixaEletronico possui os métodos logar(), sacar(), depositar() e saldo() e todas retornam uma String com a mensagem que será exibida na tela do caixa eletrônico.
* Existe uma classe chamada ContaCorrente que possui as informações da conta necessárias para executar as funcionalidades do CaixaEletronico. Essa classe faz parte da implementação e deve ser definida durante a sessão de TDD.
* As informações da classe ContaCorrente podem ser obtidas utilizando os métodos de uma interface chamada ServicoRemoto. Essa interface possui o método recuperarConta() que recupera uma conta baseada no seu número e o método persistirConta() que grava alterações, como uma mudança no saldo devido a um saque ou depósito. Não tem nenhuma implementação disponível da interface ServicoRemoto e deve ser utilizado um Mock Object para ela durante os testes.
* O método persistirConta() da interface ServicoRemoto deve ser chamado **apenas** no caso de ser feito algum saque ou depósito **com sucesso**.
* Ao executar o método saldo(), a mensagem retornada deve ser "O saldo é R$xx,xx" com o valor do saldo.
* Ao executar o método sacar(), e a execução for com sucesso, deve retornar a mensagem "Retire seu dinheiro". Se o valor sacado for maior que o saldo da conta, a classe CaixaEletronico deve retornar uma String dizendo "Saldo insuficiente".
* Ao executar o método depositar(), e a execução for com sucesso, deve retornar a mensagem "Depósito recebido com sucesso"
* Ao executar o método login(), e a execução for com sucesso, deve retornar a mensagem "Usuário Autenticado". Caso falhe, deve retornar "Não foi possível autenticar o usuário"
* Existe uma interface chamada Hardware que possui os métodos pegarNumeroDaContaCartao() para ler o número da conta do cartão para o login (retorna uma String com o número da conta), entregarDinheiro() que entrega o dinheiro no caso do saque (retorna void) e lerEnvelope() que recebe o envelope com dinheiro na operação de depósito (retorna void). Não tem nenhuma implementação disponível da interface Hardware e deve ser utilizado um Mock Object para ela durante os testes.
* Todos os metodos da interface Hardware podem lançar uma exceção dizendo que houve uma falha de funcionamento do hardware.

Deve-se criar testes também para os casos de falha, principalmente na classe Hardware que pode falhar a qualquer momento devido a um mau funcionamento.

Lembre-se de usar o TDD e ir incrementando as funcionalidades aos poucos.

Você deve entregar o código final, incluindo os testes e os mock objects criados. Coloque todo código relativo a teste em uma pasta separada.

**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
* A separação do código relativo a testes do código de produção

## ATENÇÃO - Cuidados na revisão dos trabalhos de colegas

**ATENÇÃO 1: Cuidado na Submissão do seu Trabalho**

Ao contrário do que ocorria no Curso 1, neste curso você só poderá submeter a sua atividade de revisão por pares **apenas uma única vez**. Por isso, complete sua atividade com muito cuidado, procurando fazê-la o melhor possível e só a submeta quando estiver confiante que respondeu tudo de acordo com o que foi pedido.

**ATENÇÃO 2: Revisão dos trabalhos submetidos pelos colegas**

* Solicitamos que faça a revisão do trabalho de colegas de forma a mais criteriosa possível, seguindo estritamente o que foi pedido fazer, sem extrapolar o que foi pedido no enunciado, nem aceitar problemas ou erros flagrantes.
* Identificamos alunos que, não sabemos o motivo, zeravam o trabalho sendo avaliado, sem entrar no mérito do que o responsável do trabalho tinha feito algo merecedor de zero ou não.
* Lembramos que avaliar o trabalho do colega também faz parte da sua avaliação e você tem que fazer uma avaliação a mais correta e criteriosa possível por três motivos: (i) Consolidar o seu conhecimento do assunto; (ii) Não prejudicar seu colega, atribuindo-lhe uma nota maior ou menor sem mérito; (iii) Além disso, não prejudicar sua própria nota, caso constatemos sua falta de critério na avaliação de trabalho de colega, principalmente zerando a nota sem motivo.

**ATENÇÃO 3: Cuidados na revisão dos códigos dos trabalhos de colegas**

A revisão do trabalho dos colegas deverá ser feita apenas por inspeção de código, não havendo a necessidade de executar o código recebido.

Alertamos que os códigos enviados por outros alunos poderão inadvertidamente conter elementos maliciosos com potencial de danificar sua máquina e sistema operacional, além de causar eventuais problemas de privacidade e segurança.

Dessa forma, caso deseje ainda assim executar o código do colega, recomendamos que só o faça caso o tenha inspecionado antes e saiba que é seguro.