



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 métodos 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.