

1. Toda a questão deverá utilizar o padrão MVC estudado em sala. Não é permitido utilizar `out.print` nas servlets e não é permitido utilizar scriptlet no jsp; utilize JSTL e EL. Não é permitido colocar tags HTML na controladora. Não há necessidade de verificar se o usuário está logado. Você deve desenvolver um sistema para transferir um valor de uma conta para outra usando TRANSAÇÃO. Toda exceção capturada deve ser logada utilizando `log4j`. Você irá trabalhar com o banco de dados MySQL `bd_conta`.

Esse banco possui as tabelas abaixo:

tb_conta	
id	int auto increment
numero	int
nomeCliente	varchar(200) [String]
texto	double

- (a) Crie um html que receba o número de duas contas (A e B) correntes e um valor. Transfira o valor da conta A para a conta B.
- (b) Em ContaDAO, suponha a existência do método:
 - `public Conta getConta(int numero) throws DAOException;`Escreva o código apenas do método para atualizar o saldo. Escreva o javadoc desse método explicando o que ele faz.
- (c) Crie a DAOException para retornar exceções das camadas DAO.
- (d) Crie a ConnectionFactory utilizando o Pool de conexões do tomcat. Suponha que o datasource esteja configurado com o nome `jdbc/ccDS` no `server.xml` do tomcat.
- (e) O resultado da transferência deve ser apresentado em um JSP, ou seja, coloque um boolean no request caracterizando o sucesso ou insucesso da operação. Suponha os arquivos `ContaBundle.properties` criado. Dentro desse arquivo tem duas chaves: `sucesso` e `insucesso`. Utilizando os recursos de internacionalização, exiba a mensagem de sucesso ou insucesso utilizando Resource Bundle no JSP.
- (f) Implemente:
 - i. o DAOFactory.
 - ii. o Service.
 - iii. o FrontController e o Command.

Escreva todo código necessário para compilar. Não precisa criar o `web.xml`. Não precisa fazer imports ou inserção de diretivas. Não precisa criar o JavaBean `Conta`. Suponha que o `log4j.xml` já esteja configurado corretamente.