

Desafio Qualiti Banck

Demonstre nesse desafio todas as suas habilidades de codificação. Apresente todas as boas práticas que você considera importantes da engenharia de software. Lembre-se que um sistema de qualidade é um sistema bem testado. Você poderá adicionar e estender à vontade os requisitos da aplicação, mas lembre-se da máxima K.I.S (**Keep It Simple**), ou seja, atenda aos requisitos solicitados primeiros.

Requisitos técnicos obrigatórios:

1. Estruturar a aplicação em camadas;
2. Uso de banco de dados relacional/SQL. Preferencialmente HSQLDB ou H2, entretanto pode-se usar outros como: MSQL ou Postgres;
3. Processo de build utilizando Maven;
4. Persistência utilizando JDBC ou JPA;
5. Utilizar no mínimo Java 8;
6. Utilizar na interface JSF/Primefaces ou JSP com jQuery e Ajax para o front-end;
7. Disponibilizar o código em repositório Git online (Gitlab ou Github).

Requisitos desejáveis:

1. Disponibilizar a aplicação em serviço de hospedagem online (Heroku, AWS, Digital Ocean, etc).
2. Uso de EJBs para controle transacional.
3. Uso de servidor de aplicação Wildfly ou OpenLiberty.
4. Disponibilizar a configuração do servidor de aplicação no repositório Git online junto com o código fonte.
5. Testes unitários com junit. Esse item é desejável, porem muito importante.

Requisitos funcionais:

Você foi contratado por uma *fitech* QB (Qualiti Banck) para implementar um protótipo que mistura os conceitos de rede social e banco virtual. A primeira sprint deve conter a seguintes funcionalidades.

1. Cadastro de cliente, com os seguintes dados. A princípio um cliente pode existir no QB sem necessariamente ter um conta associada.
 - a. Nome [string]
 - b. Email [string]
 - c. Senha [string]
 - d. Conta [lista]
2. Autenticação de cliente, o cliente só poderá acessar o sistema por meio de seu e-mail e senha. Para a primeira sprint um cliente poderá cadastrar e alterar os dados de outros clientes.
3. Cadastro de conta, com os seguintes dados.
 - a. Número [string]
 - b. Saldo (o saldo inicial é 0)