

Teste de Java Univalle

Nesse teste tentamos simular o que você irá encontrar no dia a dia de trabalho na Univalle.

Java

Neste teste iremos utilizar o java 8. Como referência de estrutura, utilize o arquivo .rar disponibilizado na pasta java/*.rar dentro do repositório

1) Nosso sistema irá usar um framework para importação de arquivos texto chamado Fixedformat4j.

– Insira as dependências necessárias e crie um model para o seguinte layout de produto:

Nome do Produto (20 chars);

Código do produto (10 chars);

Preço do Produto (8 casas com 3 decimais);

Data da Compra (10 chars com formatação de data yyyyMMdd).

2) Crie um serviço para inserir um produto assim como existe para Grupo de Produtos.

3) Faça as modificações abaixo:

a) No grupo de GrupoProdutosDAO crie um filtro, como achar melhor, para os grupos de produtos que começam por um parâmetro String.

b) Crie a mesma funcionalidade usando stream e lambda.

4) Muitas vezes vamos trabalhar remotamente, para ser possível o acesso do nosso ambiente de produção externamente, configure o jboss para receber requisições de fora.

5) Nossos modelos estão muito pobres, precisamos que todos eles, tenham um indicador de três estados possíveis:

– importado

– nativo

– inativo

Além disso cada modelo DEVE implementar um método de validação, faça com que estes requisitos sejam cumpridos.

FrontEnd

- **A sua tarefa**

A sua tarefa é criar uma aplicação responsiva usando AngularJS, que realize a inserção de um produto na API JAVA, para ser mostrado na tela de Produtos, usando *Service* ou *Factory*.

- **As regras**

Esses itens são obrigatórios. Não respeitá-los invalida automaticamente a sua submissão. Criamos algumas telas de exemplo, mas sinta-se a vontade para criar suas próprias.

- A aplicação deve ser Responsiva, utilize como framework o Bootstrap
- Você deve passar os dados entre os *Controllers* com um *Service* ou *Factory*
- Você deve usar algum lint para o CSS e JS
- Seu Task Runner deve ter uma tarefa de Reloading do Browser se algum arquivo for modificado
- Use um Task Runner (Grunt, Gulp, NPM scripts, etc)
- Use um Gerenciador de Pacotes (Bower, NPM, etc)
- Você deve tratar os erros
- **Esperamos que você:**

Encare esses itens como “Boas Práticas”.

- Um README, em inglês, explicando as dependências, comandos que devem ser executados
- Crie um token entre a *API* e o front-end rodados e as tarefas do seu Task Runner
- Use SCSS para o CSS
- Organize e otimize seu código e imagens
- Valide os formulários com componentes nativos do AngularJS, sem usar outras bibliotecas

Após finalizar o teste, subir em um repositório git e nos enviar o link.

Boa sorte. ;)