04

# Projeto do curso

## Transcrição

Agora que já conversamos um pouco sobre o Selenium, nesta aula conheceremos o projeto com o qual trabalharemos no curso, quais são suas funcionalidades e um pouco do seu código-fonte. Já temos uma aplicação previamente desenvolvida, já que o foco do treinamento não é entendermos a tecnologia da aplicação si, mas sim escrever os testes automatizados que a avaliarão.

> Baixe aqui o projeto inicial do curso. (https://github.com/alura-cursos/2019-selenium-java/archive/projeto_inicial.zip)

No link acima você pode fazer o download da aplicação, que deverá ser descompactado em algum diretório do seu computador e importado na IDE de sua preferência, como o Eclipse, que utilizaremos no curso, ou IntelliJ, Netbeans, entre outras. Temos uma aplicação Java utilizando o Maven - portanto, você precisará importá-la utilizando o Maven - e que segue uma estrutura padrão, utilizando também o framework Springboot.

Você não precisa conhecer o Springboot para acompanhar o curso, mas temos treinamentos sobre este framework aqui na Alura caso você tenha curiosidade.

> Formação Spring Framework na Alura! (https://cursos.alura.com.br/formacao-spring-framewor

Dentro do diretório "source/main/java" existe um pacote `alura.leilao`. Vamos trabalhar com uma aplicação de leilões, e conheceremos mais sobre ela daqui a pouco. Dentro do pacote raiz da aplicação temos a classe `LeilaoApplication`. Para rodarmos a aplicação com o Springboot, é necessário executar essa classe. No Eclipse, fazemos isso clicando com o botão direito sobre a classe em seguida em "Run As > Java Application".

Ao executarmos como uma aplicação Java, vão aparecer algumas informações no console do Eclipse. A aplicação tem um servidor embutido, que é o Tomcat, e será executada na porta `8080`.

Após inicializado, podemos abrir um navegador e acessar a aplicação por meio do endereço http://localhost:8080 (http://localhost:8080). Ela simula um sistema de leilões, no qual as pessoas podem cadastrar leilões ou dar lances para comprar produtos - claro, caso consigam o maior lance. Ao entrarmos na aplicação, somos direcionados à página que mostra os leilões cadastrados.

Ao executarmos a classe `LeilaoApplication`, já populamos o banco de dados com algumas informações - dessa forma, não teremos que nos preocupar em cadastrar várias informações toda vez que rodarmos o projeto. Na tela principal, temos acesso somente à lista de leilões, onde atualmente temos dois leilões cadastrados. Um leilão precisa possuir um nome, uma data de abertura, o valor do lance inicial e o usuário que criou o leilão.

Para cadastrar um leilão ou dar um lance, é necessário fazer login na aplicação. No canto superior direito da tela temos um botão "Entrar" que nos leva a um formulário de login. Existe um usuário previamente cadastrado com o nome "fulano" e a senha "pass". Ao preechermos e submetermos as informações clicando em "Login", seremos levados de volta à tela de leilão, mas com algumas opções a mais.

VOLTAR AO TOPO

A primeira delas é o botão "novo leilão", que permite cadastrar um novo leilão. Em nosso teste, preencheremos o campo "Nome" com "Iphone 10"; o "Valor inicial" com `4000` ; a "Data de abertura" com "20/11/2020"; e o "Usuário" é preenchido automaticamnete com o nome do usuário logado no sistema. Clicando em "Salvar", o leilão será cadastrado no sistema e aparecerá na listagem.

Como fomos nós que cadastramos o leilão, não podemos dar um lance, somente editá-lo. O botão "editar" nos leva de volta ao formulário e nos permite, por exemplo, alterar o "Nome" (que substituiremos por "Iphone 5"). Ao salvarmos, as informações serão atualizadas na listagem.

Se for um leilão criado por outro usuário, poderemos dar lances. Como exemplo, temos um leilão criado pelo usuário "beltrano". Ao clicarmos em "dar lance", somos levados a uma página que mostra todos os lances, mas atualmente não há nenhum. Preencheremos o campo "Novo lance" com o valor "1000" e clicaremos em "Dar lance!". A página será atualizada e o novo lance será exibido na listagem. Se preenchermos e submetermos um novo lance, por exemplo "1010", receberemos um erro - afinal, não deve ser possível que um mesmo usuário dê dois lances na sequência. Sendo assim, é necessário esperar outro usuário dê um lance para que "fulano" crie um lance maior.

Temos então uma aplicação bem simples, com cadastro de leilões, tela para editar informações de um leilão que criamos, tela para dar lances em leilões de outros usuários e algumas regras de negócio em relação ao valor mínimo, data e lances. Por exemplo, um usuário só pode dar cinco lances em um determinado leilão, e não pode dar dois lances seguidos.

No pacote `alura.leilao` , encontramos outros pacotes com as classes, controladores, classes de modelo (utilizando JPA) e assim por diante - ou seja, estamos trabalhando com uma aplicação Java tradicional, [illegible]s são

encontradas no diretório "src/main/resources/templates", e consistem em páginas HTMLe utilizando Thymeleaf.

Não é necessário conhecer essas tecnologias, ainda que conhecer o básico delas possa ser de alguma ajuda, já que não entraremos em detalhes sobre o código fonte, como as funcionalidades foram implementadas e assim por diante. O foco do curso realmente é nos testes automatizados utilizando o Selenium. Obviamente, em alguns momentos, será necessário passarmos por algumas classes e páginas do projeto, mas não perderemos muito tempo nisso.

Após termos baixado e importado o projeto no computador, podemos partir para a próxima aula, na qual escreveremos o primeiro teste automatizado utilizando o Selenium!

VOLTAR AO TOPO