02

Inserindo códigos SQL

Transcrição

Nós fomos contratados pela **Alura Shows**, com os objetivos de realizarmos teste na aplicação que eles desenvolveram utilizando ***Spring MVC***. O intuito é encontrar possíveis vulnerabilidades de segurança que podem existir, e caso a gente encontre alguma, a nossa tarefa será corrigi-la e torná-la mais segura.

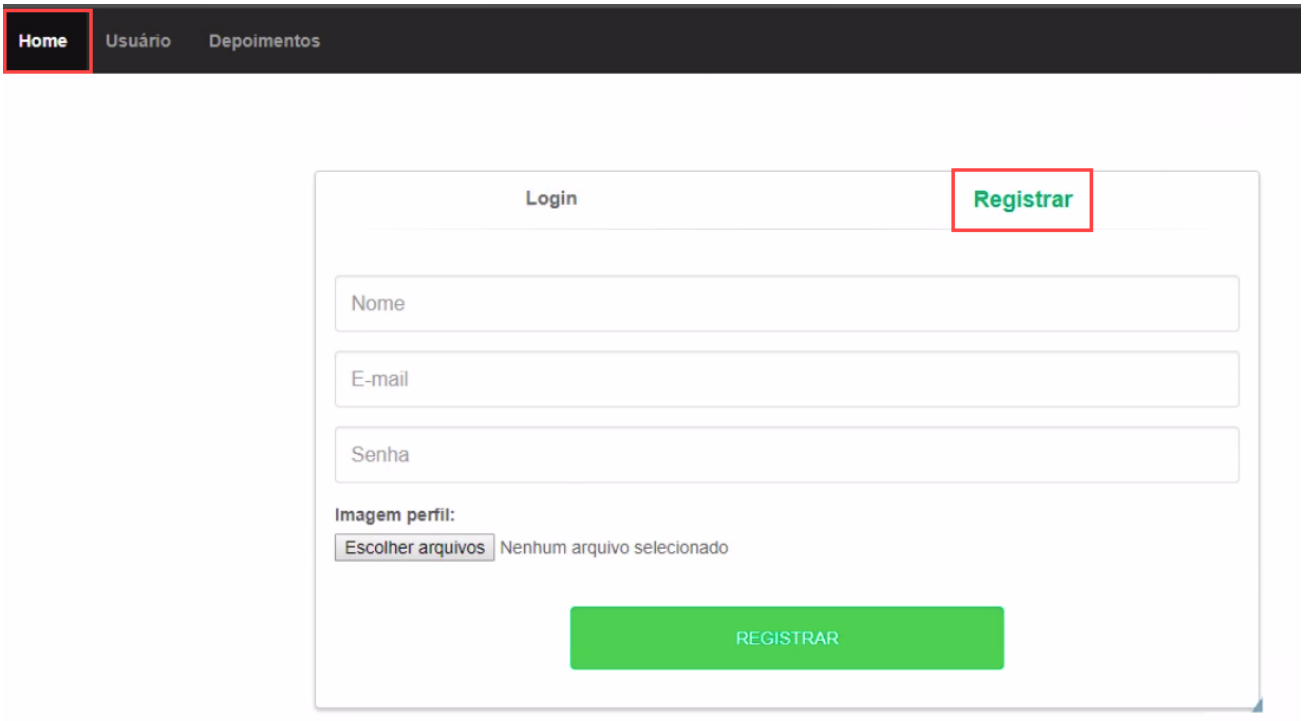
No primeiro dia de reunião com os diretores da Alura Shows, nos foi passado os arquivos do projeto para começarmos a gerar os testes.

Com o projeto no Eclipse, para iniciarmos os testes, é necessário acessarmos a aplicação. Podemos arrastar com o mouse o projeto do ***Package Explorer***, e soltá-lo no servidor configurado na aba "Servers". Agora basta pressionarmos o ***Start the Server** para que a aplicação inicie.

Após o ***Tomcat*** ser inicializado, acessaremos no navegador o endereço `localhost:8080/alura-shows/` . Seremos redirecionados para a página inicial da aplicação, que tem informações dos próximos eventos.

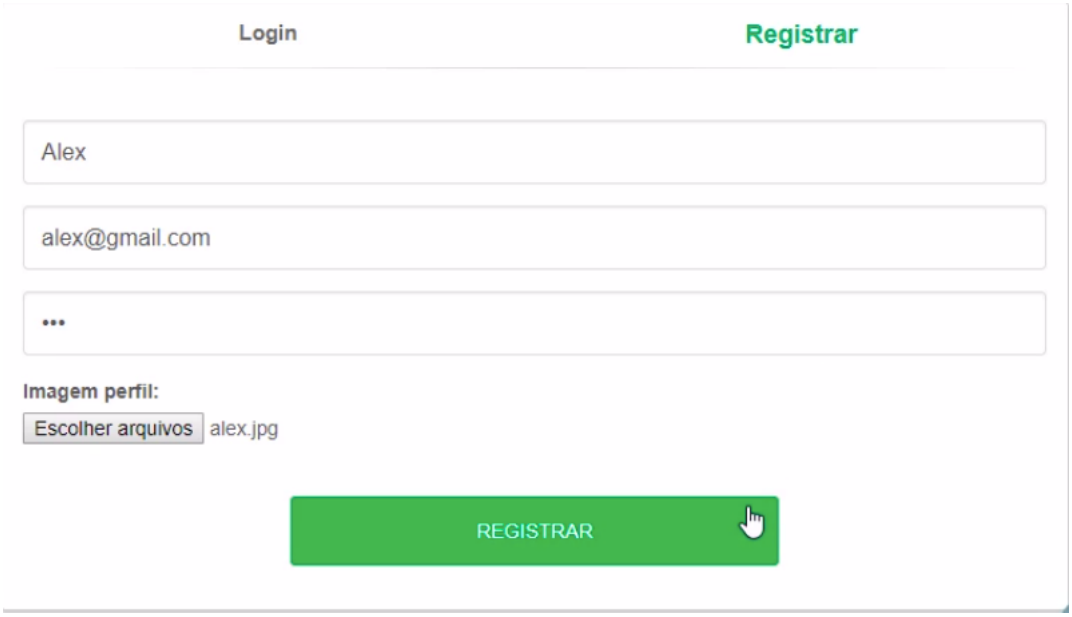
O objetivo da aplicação, é que os usuários interajam com o sistema para visualizar as datas dos próximos shows, comprar ingressos e assim por diante. Mas para isso, em algum momento o usuário terá que fazer um cadastro.

Para se cadastrar, o usuário tem que ir em "Usuário > Registrar":

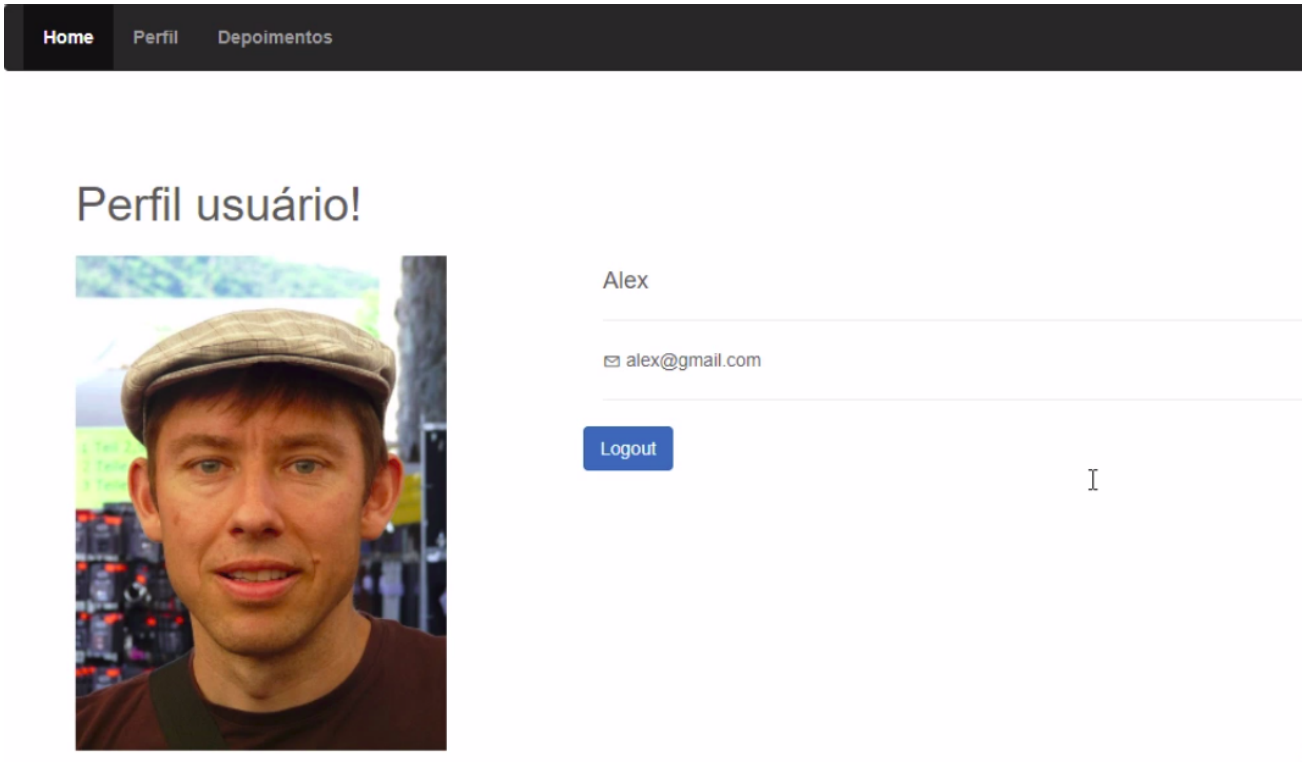


Cadastraremos o primeiro usuário que se chama **Alex**. Colocaremos as seguintes informações no formulário de cadastro:

- **Nome:** `Alex`
- **E-mail:** `alex@gmail.com`
- **Senha:** `123`
- **Imagem perfil:** `alex.jpg`



Ao clicarmos em **Registrar**, as informações do usuário **Alex** serão armazenadas no banco de dados, em seguida seremos redirecionados para uma página que mostrará as informações do usuário. Após isso podemos clicar em "Logout".



A **Alura Shows** está ficando cada vez mais famosa, e outros usuário também querem se cadastrar no sistema. Cadastraremos outro usuário que se chama **Fernando**. Colocaremos as seguintes informações no formulário de cadastro:

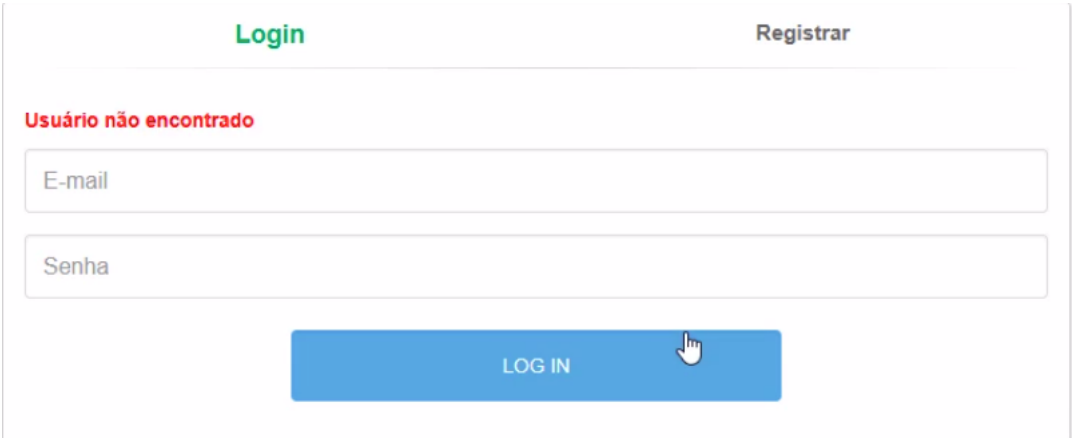
- **Nome:** `Fernando`
- **E-mail:** `fernando@gmail.com`
- **Senha:** `456`

- **Imagem perfil:** `fernando.jpg`

Temos os usuários **Alex** e **Fernando** cadastrados. Em algum momento, esses usuários terão que fazer uma autenticação no sistema. Vamos testar a autenticação.

Clicaremos em "Logout" para sair do perfil do **Fernando**. Em ***Login***, no campo **E-mail** colocaremos `alex@gmail.com e no campo **Senha** colocaremos 123`. Como as informações estão corretas, é esperado que a autenticação seja aceita.

A autenticação funcionou! Mas, se em vez de usarmos a senha como `123` e colocarmos `0123456789`, o que aconteceria? Receberemos a mensagem de `Usuário não encontrado`



Isso mostra que a validação da etapa de *login* está sendo feita, não permitindo que usuários com informações erradas acessem a aplicação.

Mas será que os desenvolvedores se preocuparam com os caracteres passados no formulário? Por exemplo, se colocarmos no campo **E-mail** `alex@gmail.com e em **Senha** o caractere '` (aspas simples), o que aconteceria?

Recebemos uma ***Exception*** do sistema. A mensagem de erro emitida pela *Exception* é:

> *You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version*

Algo como:

> Você tem um erro na sua sintaxe SQL. Olhe o manual correspondente à sua versão do MySQL

Isso significa que os desenvolvedores da **Alura Shows** não estão fazendo a verificação de caracteres. Isso pode se tornar uma grande vulnerabilidade da aplicação.

Veremos o que podemos fazer em relação a essa injeção de código SQL.