



## Verificando o login

### Transcrição

Bem vindo de volta! Na aula anterior nós criamos uma página de login, mas nossa aplicação ainda não reconhece se o usuário fez ou não login - todo mundo continua tendo acesso à página `ListaEmpresas`, e queremos restringir os recursos e funcionalidades da nossa aplicação somente para quem fizer login.

Até o momento, somente aproveitamos os recursos do nosso framework para criarmos as ações `LoginForm` (que chama o formulário) e `Login` (que é o início da lógica de autenticação).

Agora precisamos que nossa ação `Login` verifique se o usuário existe ou não, da mesma forma que outros sites (como a [Alura](https://cursos.alura.com.br/dashboard/) (<https://cursos.alura.com.br/dashboard/>) ou o Facebook) fazem.

A ideia é utilizarmos o nosso banco (que representa a nossa camada de persistência). Faremos isso escrevendo, na classe `Login`, o código `Banco banco = new Banco()`. Além disso, devemos importar a classe `Banco` com "Ctrl + Shift + O".

Agora criaremos um método `existeUsuario()` que recebe `login` e `senha`. Queremos que esse método devolva um `usuario` se esse usuário existio. Do contrário, queremos que ele devolva `null`. Assim, teremos:

```
Usuario usuario = banco.existeUsuario(login, senha);
```

[COPIAR CÓDIGO](#)

Poderíamos ter feito uma implementação diferente, mas vamos trabalhar dessa maneira. Agora usaremos o Eclipse para adicionar esse método no `Banco`. Faremos isso clicando no ícone de erro ou sobre `existeUsuario()` e em seguida em "Create method 'existeUsuario(String, String)' in type 'Banco'":

```
public Usuario existeUsuario(String login, String senha) {  
    return null;  
}
```

[COPIAR CÓDIGO](#)

Para verificarmos no banco de dados se o usuário existe ou não, criaremos um laço `for()`: para cada `usuario` em `listaUsuarios`, queremos saber se esse `usuario` possui as credenciais `login` e `senha`. Para fazermos essa verificação, utilizaremos o método `ehIgual()` que preparamos anteriormente.

```
public Usuario existeUsuario(String login, String senha) {  
    for(Usuario usuario : listaUsuarios) {  
        if(u.ehIgual(login, senha)) {  
            return usuario;  
        }  
    }  
    return null;  
}
```

[COPIAR CÓDIGO](#)

Se essa verificação for verdadeira, a aplicação deve retornar o `usuario`. Se não, deve devolver `null`. Vamos testar essa funcionalidade na classe `Login`, fazendo com que, se `usuario` for diferente de `null`, o console imprima "Usuario existe":

```
Banco banco = new Banco();  
Usuario usuario = banco.existeUsuario(login, senha);  
  
if(usuario !=null ) {  
    System.out.println("Usuario existe");  
}
```

[COPIAR CÓDIGO](#)

Aparentemente tudo está funcionando como deveria, correto? Agora, se o usuário existe, queremos que ele seja redirecionado para `ListaEmpresas`. Mas e se ele não existe? Queremos que ele retorne ao formulário de login! Vamos escrever isso:

```
if(usuario !=null) {  
    System.out.println("Usuario existe");  
    return "redirect:entrada?acao=ListaEmpresas";  
} else {  
    return "redirect:entrada?acao=LoginForm";  
}
```

[COPIAR CÓDIGO](#)

Dessa vez, se tentarmos fazer login com um usuário ou senha que não existe, seremos redirecionados de volta ao formulário. Portanto, já conseguimos verificar se o usuário existe no banco ou não, e tomar uma atitude a partir disso. Porém, nossa motivação inicial é restringir o acesso às informações para usuários que fizeram login, e isso ainda não está funcionando.

Vamos trabalhar nisso no próximo vídeo. Até lá!

