

## Trabalhando com o GoogleRecaptcha

### Transcrição

Por meio do *Burp Suite*, o usuário **Alex** conseguiu efetuar um ataque automatizado de **força bruta** e o resultado foi a descoberta da senha cadastrada pela **Ana** na aplicação da Alura Shows. Para evitar que esse tipo de ataque seja realizado, usaremos o *reCaptcha* da Google.

Acessaremos a página do *reCaptcha* (<http://www.google.com/recaptcha>). Na página inicial veremos o lema:

Tough on bots. Easy on humans.

Em tradução livre:

Difícil em bots. Fácil em humanos.

A ideia é colocarmos o *reCaptcha* no formulário de *Login* para que só seja validado por um clique de usuário humano. Para utilizarmos o *reCaptcha* precisaremos de um conta no *Gmail* (<http://www.google.com/gmail/>). Clicaremos na em *Get reCaptcha*.

Em *Label* colocaremos o nome do *reCaptcha* que queremos registrar, chamaremos de `alura-shows`. Em *Choose the type of reCAPTCHA*, marcaremos a opção "reCAPTCHA V2". Na seção *Domain*, colocaremos o domínio que iremos vincular, como estamos rodando o projeto localmente colocaremos o endereço IP do *localhost* `127.0.0.1` e do Windows `192.168.121.171` (Seu computador terá o IP diferente). No final basta aceitar os termos da licença e clicar em "Registrar".

O Google mostrará um guia de como configurar o *reCAPTCHA* no projeto. O primeiro passo é importar a Tag com código JavaScript:

```
<script src='https://www.google.com/recaptcha/api.js'></script>
```

No Eclipse, pararemos o Tomcat e abriremos o arquivo `rodape.jsp`. Colaremos a Tag com código JavaScript após todas as outras Tags `<script>`.

```
<!-- FOOTER -->
<footer class="">
  <div class="container">
    <p class="text-muted">&copy; 2017 Alura Shows, Inc</p>
  </div>
</footer>

</div>

<script
  src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script src="<c:url value="/resources/js/jquery-ui.js"/>"></script>
<script src="<c:url value="/resources/js/bootstrap.min.js"/>"></script>
<script src="<c:url value="/resources/js/config.js"/>"></script>
```

```
</body>  
</html>
```

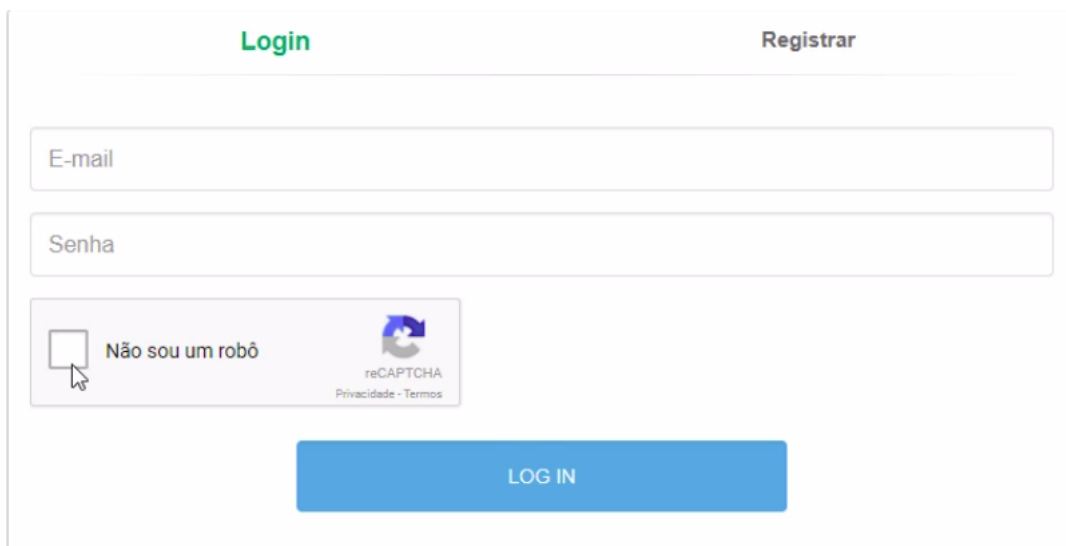
O segundo passo é colocar a Tag que contém de fato o campo de *reCAPTCHA* na aplicação. A Tag está da seguinte maneira:

```
<div class="g-recaptcha" data-sitekey="6LddPTUUAaaaaafibLDLxpz7iQC5AknyHrs-T-A1"></div>
```

Abriremos o arquivo `usuario.jsp` e colocaremos a Tag `<div>` do *reCAPTCHA* abaixo do campo da senha no formulário. O formulário ficará da seguinte maneira:

```
<form id="login-form" action="{s:mvcUrl('UC#login')}.build()}"  
  method="get" role="form" style="display: block;">  
  <div class="form-group">  
    <input type="text" name="email" id="email" tabindex="1"  
      class="form-control" placeholder="E-mail" value="" required="required">  
  </div>  
  <div class="form-group">  
    <input type="password" name="senha" id="senha" tabindex="2"  
      class="form-control" placeholder="Senha" required="required">  
  </div>  
  
  <div class="g-recaptcha" data-sitekey="6LddPTUUAaaaaafibLDLxpz7iQC5AknyHrs-T-A1"></div>  
  
  <div class="form-group" style="margin-top: 3%">  
    <div class="row">  
      <div class="col-sm-6 col-sm-offset-3">  
        <input type="submit" name="login-submit" id="login-submit"  
          tabindex="4" class="form-control btn btn-login"  
          value="Log In">  
      </div>  
    </div>  
  </div>  
</form>
```

Reiniciaremos o Tomcat e acessaremos a aplicação no navegador. Como cadastramos no *reCAPTCHA* os endereços IPs para a utilização, precisamos colocar no navegador o endereço `127.0.0.1:8080/alura-shows/usuario`.



Agora veremos como validar o clique do usuário.