

## Mão à obra: Autenticação com o Google Recaptcha

Uma vez que temos a informação do atributo **success**, vamos retornar essa informação para o método `login` na classe `UsuarioController` para definirmos se iremos continuar com o login do usuário ou se iremos colocar uma mensagem para o usuário confirmar que é humano. Vá até a classe `GoogleWebClient` e retorno o valor do método `isSuccess()`

```
public boolean verifica(String recaptcha) throws IOException{
    Call<Resposta> token = new RetrofitInicializador().getGoogleService().enviaToken(SECRET, recaptcha);
    return token.execute().body().isSuccess();
}
```

Como estamos trabalhando com o Spring, vamos pedir para o Spring fazer o gerenciamento do estado desse objeto, anotando essa classe com `@Component`:

```
@Component
public class GoogleWebClient {

    private static final String SECRET = "[sua chave de segredo]";

    public boolean verifica(String recaptcha) throws IOException{
        Call<Resposta> token = new RetrofitInicializador().getGoogleService().enviaToken(SECRET, recaptcha);
        return token.execute().body().isSuccess();
    }
}
```

Feito isso, vamos voltar na classe `UsuarioController` e pedir para o Spring fazer a injeção através da anotação `@Autowired`:

```
@Autowired
private GoogleWebClient cliente;
```

Com isso, não precisamos mais instanciar a classe `GoogleWebClient`:

```
boolean verificaRecaptcha = cliente.verifica(recaptcha);
```

Podemos refatorar o código e colocar todo o conteúdo da lógica de login do usuário em um método chamado `procuraUsuario`:

```
private String procuraUsuario(Usuario usuario, RedirectAttributes redirect,
                               Model model, HttpSession session) {

    Usuario usuarioRetornado = dao.procuraUsuario(usuario);
    model.addAttribute("usuario", usuarioRetornado);
    if (usuarioRetornado == null) {
        redirect.addFlashAttribute("mensagem", "Usuário não encontrado");
```

```
        return "redirect:/usuario";  
    }  
  
    session.setAttribute("usuario", usuarioRetornado);  
    return "usuarioLogado";  
}
```

Volte ao método login para que possamos fazer a validação do retorno do método verifica da classe GoogleWebClient. Caso o método retorne true, é porque a validação do recaptcha foi aceita e seguiremos com o login do usuário, do contrário, o clique foi inválido e retornamos uma mensagem para o usuário para ele provar que é humano:

```
if(verificaRecaptcha){  
    return procuraUsuario(usuario, redirect, model, session);  
}  
  
redirect.addFlashAttribute("mensagem", "Por favor, comprove que você é humano!");  
return "redirect:/usuario";
```

Volte na aplicação e tente fazer o login novamente com o usuário Ana, com o e-mail `ana@gmail.com`, senha `789` e clique no botão do Recaptcha. Qual é o resultado? Você é redirecionado para a parte do perfil do usuário? Na sequência faça o teste sem clicar no botão do Recaptcha, o que acontece?