

09

Para saber mais: Timeout da sessão

Imagina que você acessou hoje a nossa aplicação em 3 computadores diferentes. Isso significa que o servidor Tomcat criou no mínimo 3 objetos `HttpSession` para você só nesse dia, pois cada navegador é identificado por um ID de sessão diferente.

Pior ainda, quando você fecha o navegador, esse cookie com ID será automaticamente removido, pois trata-se de um **session cookie** que só vive enquanto o navegador está aberto (é um cookie transiente).

Vamos pensar que você adora fechar e abrir o navegador e fez isso 5 vezes em cada computador. Ou seja, fazendo o cálculo:

$$3 \text{ computadores} \times 5 \text{ vezes navegador reaberto} = 15 \text{ objetos HttpSession}$$

Foram então 15 objetos `HttpSession` só para você nesse dia. Imagine isso para uma aplicação com muitos acessos, por exemplo 1000 usuários por dia! Seriam 15000 objetos `HttpSession` em memória, em um dia!

Realmente, seriam milhares de objetos em memória, mas saiba que o objeto `HttpSession` tem um ciclo de vida. Isso significa que ele será criado mas também será destruído.

Acontece que a `HttpSession` tem um *timeout* associado. Se você não usa a nossa aplicação por um determinado tempo, o Tomcat automaticamente remove o objeto `HttpSession` da memória. O padrão do Tomcat 9 é de 30 minutos, ou seja, se você não usar a aplicação por 30 min, você será deslogado!!

Talvez você ache os 30 minutos pouco ou muito tempo, mas saiba que isso é configurável através do nosso `web.xml`, basta colocar o seguinte trecho:

```
<session-config>
  <!-- 10 min -->
  <session-timeout>10</session-timeout>
</session-config>
```

Só reforçando: é um *timeout de desuso*. No caso acima, o Tomcat só removerá a sessão se o usuário não ficar ativo por 10 minutos.

Obs: Você já deve ter percebido que a Alura não perde o seu login quando você reabre o navegador. Isto acontece pois a Alura não usa o cookie padrão e sim cria o seu próprio cookie persistente (e não transiente).