



# Programação para Internet

---

## Módulo 8

### Sessões em PHP

Prof. Dr. Daniel A. Furtado - FACOM/UFU

Conteúdo protegido por direito autoral, nos termos da Lei nº 9 610/98

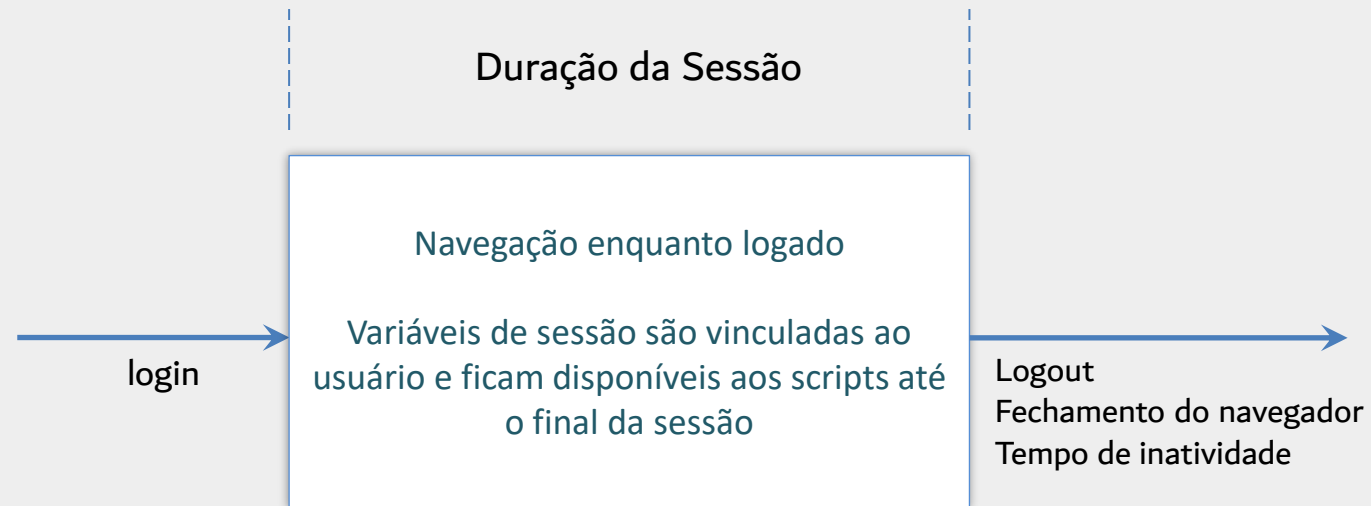
A cópia, reprodução ou apropriação deste material, total ou parcialmente, é proibida pelo autor

# Introdução

- Necessidade de armazenamento temporário de dados que possam ser compartilhados com vários scripts PHP
- Dados mantidos ao longo de uma sessão de acesso
- Dados automaticamente excluídos ao término da sessão
- Uma solução é utilizar o conceito de **variáveis de sessão** do PHP

Por exemplo, em um sistema web de acesso restrito é desejável armazenar temporariamente alguns dados do usuário logo após o seu *login*, como o ID do usuário, e-mail, data e hora do *login*, dados do navegador, etc. A ideia é compartilhar esses dados com todos os scripts da parte restrita enquanto o usuário estiver *logado* no sistema (e sem salvar os dados em cookies no computador do usuário);

# Conceito de Sessão



# Exemplo

Script teste-sessao1.php

```
1 <?php
2
3 // inicializa a sessão
4 session_start();
5
6 // nome e e-mail recebidos de formulário
7 $nome = $_POST['nome'] ?? '';
8 $email = $_POST['email'] ?? '';
9
10 // armazena os valores em variáveis de sessão
11 $_SESSION['nome'] = $nome;
12 $_SESSION['email'] = $email;
13
14 ?>
15 <!DOCTYPE html>
16 <html><body>
17     <h2>Variáveis de sessão definidas!</h2>
18     <h3>
19         <a href="teste-sessao2.php">
20             Acessar outro script PHP
21         </a>
22     </h3>
23 </body></html>
```

Script teste-sessao2.php

```
1 <?php
2
3 // abre a sessão criada anteriormente
4 session_start();
5
6 ?>
7 <!DOCTYPE html>
8 <html><body>
9
10 <?php
11
12 // acessa as variáveis de sessão definidas anteriormente
13 // no script teste-sessao1.php
14 $nome = $_SESSION['nome'];
15 $email = $_SESSION['email'];
16
17 echo "Bem vindo, $nome <br>";
18 echo "Seu e-mail é $email <br>";
19 echo "Dados guardados em var. de sessão pela pág. ant.";
20
21 ?>
22
23 </body>
24 </html>
```

# Local de Armazenamento das Variáveis

- As variáveis de sessão são **armazenadas no servidor** (e não no computador do usuário, como em *cookies*)
- Apenas o **código da sessão** (ID) é armazenado no computador do usuário por meio de um *cookie* (mas não os dados em si)

# Resposta HTTP do Script teste-sessao1.php

localhost/ExSession/Ex01-session-teste-sessao1.php

Variáveis de sessão definidas!

[Acessar outro script PHP](#)

Elements Console Sources Network Performance

Filter ☐ Invert ☐ Hide data URLs

All Fetch/XHR JS CSS Img Media Font Doc WS Wasm Manifest Other

☐ Has blocked cookies ☐ Blocked Requests ☐ 3rd-party requests

20 ms 40 ms 60 ms 80 ms 100 ms

Name teste-sessao1.php

Headers Payload Preview Response Initiator Timing

**Response Headers** View source

Cache-Control: no-store, no-cache, must-revalidate

Connection: Keep-Alive

Content-Length: 187

Content-Type: text/html; charset=UTF-8

Date: Thu, 28 Jul 2022 14:43:42 GMT

Expires: Thu, 19 Nov 1981 08:52:00 GMT

Keep-Alive: timeout=5, max=100

Pragma: no-cache

Server: Apache/2.4.51 (Win64) PHP/8.1.0

**Set-Cookie: PHPSESSID=k23r99pt87gl6m8p55m3of1ulh; path=/**

1 requests 594 B tra

Quando a função `session_start()` é chamada pela primeira vez (no script teste-sessao1.php), uma nova sessão é criada e será gerado um código de identificação para a sessão.

O servidor web/PHP responderá a requisição HTTP com um *response header* do tipo `set-cookie`, instruindo o navegador a criar o respectivo *cookie* no computador do usuário para armazenar o código da sessão.

# Local de Armazenamento das Variáveis

- Nas próximas requisições, o *cookie* contendo o ID da sessão é enviado de volta ao servidor e permitirá o carregamento adequado das variáveis de sessão
- Portanto, o *cookie* não armazena nenhum dado de variável de sessão em si, mas apenas o ID que permitirá que os dados da sessão sejam localizados no servidor

# Cabeçalhos da Requisição HTTP - teste-sessao2.php

The screenshot shows a web browser with two tabs. The active tab is at `http://localhost/ExSession/Ex01-session-test/teste-sessao2.php`. The page content on the left reads: "Bem vindo, ciclano", "Seu e-mail é fulano@mail.com", and "Dados guardados em var. de sessão pela pág. ant.".

The browser's developer tools are open to the Network tab. A list of requests is shown, with `teste-sessao2.php` selected. The request headers for this file are displayed in the right pane:

- `X-Powered-By: PHP/8.1.0`
- Request Headers** (expanded):
  - `Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9`
  - `Accept-Encoding: gzip, deflate, br`
  - `Accept-Language: pt-BR,pt;q=0.9,en-US;q=0.8,en;q=0.7`
  - `Cache-Control: no-cache`
  - `Connection: keep-alive`
  - Cookie: PHPSESSID=k23r99pt87gl6m8p55m3of1ulh**
  - `Host: localhost`

At the bottom of the network panel, it indicates "1 requests" and "511 B tra".

No exemplo apresentado, quando o segundo arquivo PHP foi requisitado (`teste-sessao2.php`), o cookie contendo o ID da sessão é enviado de volta ao servidor, e permitirá que as variáveis de sessão adequadas sejam carregadas e utilizadas no script PHP.



# Tempo de Permanência dos Dados da Sessão

- A sessão será perdida e as variáveis de sessão poderão ser excluídas nas seguintes situações:
  1. O usuário fechar o navegador. Neste caso, o cookie da sessão é excluído pelo navegador e os dados não poderão ser recuperados no servidor
  2. Tempo de inatividade maior que o tempo definido em ***session.gc\_maxlifetime***, no arquivo de configuração do PHP (php.ini). Neste caso, o *garbage collector* do PHP tratará o conteúdo das variáveis de sessão como lixo, e poderá excluí-las
  3. O desenvolvedor encerra a sessão chamando os métodos **session\_unset()** e **session\_destroy()**

# Funções e Arrays Relacionados a Sessões

Função/Variável PHP	Descrição
<code>session_start()</code>	Função para criar uma nova sessão (caso não exista) ou abrir uma sessão existente. <b>OBS:</b> Deve ser chamada no início do script PHP, antes de qualquer código HTML ou de qualquer 'echo'
<code>\$_SESSION</code>	Array super global. Permite criar e acessar as variáveis de sessão.
<code>session_unset()</code>	Remove todas as variáveis da sessão iniciada com <code>session_start</code> (apenas apaga as variáveis de <code>\$_SESSION</code> ).
<code>session_destroy()</code>	Finaliza a sessão removendo os dados efetivamente armazenados no servidor relativos à sessão.

# Resumo do Funcionamento de Sessões no PHP

1. O ID da sessão é enviado para o computador do usuário no momento em que a sessão é criada (`session_start()` é chamada pela 1ª vez)
2. Tal ID é armazenado no computador do usuário em um *cookie* (chamado, por padrão, de PHPSESSID)
3. O *cookie* é enviado pelo navegador ao servidor toda vez que uma nova requisição HTTP é realizada
4. O servidor web utiliza o *cookie*, que contém o ID da sessão, para localizar as variáveis de sessão do usuário
5. Tais dados são carregados pelo PHP no *array* super global `$_SESSION` quando a função `session_start()` for chamada novamente

# Referências

- <https://www.php.net/docs.php>
- NIXON, R. ***Learning PHP, MySQL & JavaScript:*** With jQuery, CSS & HTML5. 5. ed. O'Reilly Media, 2018