

# Sessões

## Relatório

Gabriel Oliveira

**Controle de sessões com PHP**

# Sessões

## Relatório

# **Controle de sessões com PHP**

Relatório sobre controle de sessões  
utilizando a linguagem PHP.  
Sob a coordenação do Boot Santos.

Gabriel Oliveira Souza

Julho  
08/2021

## Sumário

1	<b>Resumo .....</b>	<b>1</b>
2	<b>O que é?.....</b>	<b>1</b>
3	<b>Como funciona?.....</b>	<b>1</b>
4	<b>Como é gerenciado no back-end?.....</b>	<b>1</b>

## 1 **Resumo**

O objetivo deste relatório é realizar um estudo mais aprofundado referente ao tema de sessões utilizando a linguagem php. Contudo, neste relatório utilizaremos imagens e texto para dar mais valorização ao conteúdo descrito.

## 2 **O que é?**

Sessões em PHP é basicamente um fluxo de armazenamento de dados individuais através de uma variável global chamada `$_SESSION`, sendo armazenada no navegador do usuário através de cookies de sessão.

## 3 **Como funciona?**

Quando a sessão é iniciada através do comando `session_start()` no código PHP, inicia-se todo um processo de armazenamento de informações. Portanto, a variável super-global `$_SESSION` será preenchida com os dados de sessão

## 4 **Como é gerenciado no back-end?**

O código é feito de uma maneira bem simples e fácil de ser entendido. Segue o exemplo abaixo de uma tela de login simplificada.

Realizaremos a criação de 2 arquivos, o primeiro chamado **login\_simples.php**, onde será feito o envio do nosso formulário pelo método POST e o arquivo **login\_s.php** onde será armazenado em nossa variável global o e-mail do usuário. Com isso, podemos demonstrar de forma melhor como o conceito de sessão funciona na prática.

## login\_simples.php:

```
boot > login_simples.php
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Login</title>
6  </head>
7
8  <!-- Corpo do código HTML -->
9  <body>
10
11      <div>
12          <h1>Login Simples &D</h1>
13      </div>
14
15      <!-- Ações ao clicar no botão de enviar.
16          1. o formulário terá uma ação juntamente com o código do arquivo login_s.php
17          2. será enviado no método POST
18      -->
19      <form action="login_s.php" method="POST">
20
21          <div>
22              <!-- Input de email -->
23              <label>Email:</label>
24              <input type="email" name="email" id="email">
25          </div>
26
27          <div>
28              <!-- Input de senha -->
29              <label>Senha:</label>
30              <input type="password" name="password" id="password">
31          </div>
32
33          <!-- Botão de envio do formulário -->
34          <button>Enviar</button>
35      </form>
36
37  </body>
38  </html>
```

## login\_s.php:

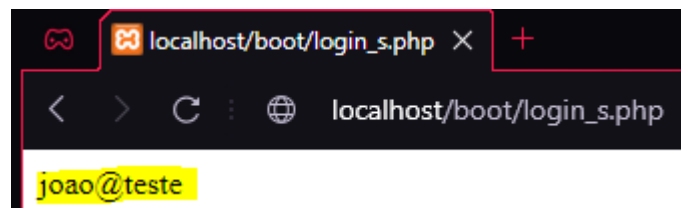
```
boot > login_s.php
1  <?php
2
3      // Instanciando e incluindo os dados passando pelo usuário nas variáveis
4      $email      = isset($_POST['email'])?$_POST['email']:'0';
5      $senha      = isset($_POST['password'])?$_POST['password']:'0';
6
7      // Iniciando sessão em php
8      session_start()
9
10     // Variável global recebendo o e-mail do usuário
11     $_SESSION['email'] = $email;
12
13     // Mostrando na tela o resultado na variável global
14     echo $_SESSION['email'];
15  ?>
```

Dando sequência, a página de login será enviados dois tipos de dados, o e-mail do usuário e a senha do usuário. Utilizaremos o e-mail fictício **joao@teste** e a senha **1234** para realizar o exemplo.



The screenshot shows a web browser window with the address bar displaying 'localhost/boot/login\_simples.php'. The page content includes a title 'Login Simples =D' in a large, bold, serif font. Below the title, there is a login form with two input fields. The first field is labeled 'Email:' and contains the text 'joao@teste'. The second field is labeled 'Senha:' and contains four dots, indicating a password. Below these fields is a button labeled 'Enviar'.

Na imagem acima, podemos ver que inserimos os dados e ao clicar no botão de enviar, obtemos o seguinte resultado.

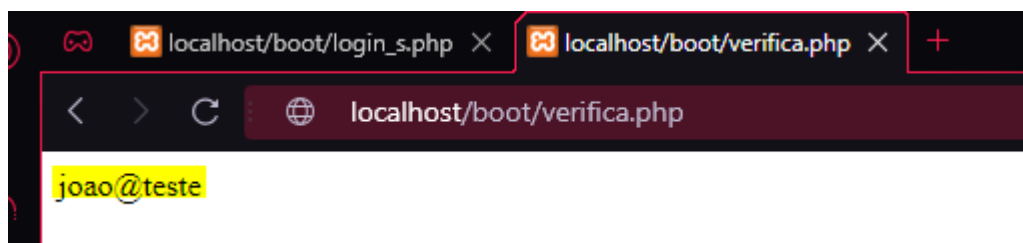


No código **login\_s.php**, utilizamos o comando **echo** para ser mostrado no nosso navegador, o conteúdo da variável de sessão. Foi criado um novo arquivo chamado **verifica.php**, sua função é basicamente mostrar na tela a nossa variável de sessão. Ao abrirmos mais uma aba e executar o arquivo **verifica.php** no mesmo navegador, percebemos que não é alterada essa variável.

### verifica.php:

```
boot > verifica.php
1 <?php
2 // Iniciando sessões em php
3 session_start();
4
5 // Mostrando na tela o resultado na váriavel global
6 echo $_SESSION['email'];
7
```

Exemplo:



Portanto, considerando as informações supracitadas acima. O controle de sessão é uma forma de gerenciamento de acesso por parte da aplicação. A forma mais básica para gerenciar o controle de sessão é a utilização do ID, em que, tem como finalidade ser utilizado como um identificador de sessão.

A finalidade do mesmo, visa verificar através das trocas de verificações do navegador junto ao servidor em cada nova navegação. Portanto, tornando o sessions IDs um token de verificação para controle.