



# **Autenticação em PHP**

## **Mecanismos de segurança**

### **Sessions e Cookies**

# Mecanismos de segurança

**Cookies** e **Sessões** são cada vez mais implementados em sistemas devido a grande necessidade de **segurança** e **personalização** de informações nas relações entre serviços e usuários.

Estes podem ser utilizados na **autenticação** de usuários, exibição de anúncios e **conteúdo direcionado**, personalização de páginas e sistemas de compras online.

Uma **sessão** permanece ativa somente enquanto o usuário permanecer no site.

Essas sessões são em geral usadas em sistemas de autenticação e consistem em arquivos com variáveis de ambientes registradas e armazenadas no servidor.

Iniciar uma sessão em PHP:

```
session_start();
```

Para registrar variáveis da sessão que poderão ser acessadas em todas as páginas do site.

```
$_SESSION["nome_variavel"] = "valor";
```

OU

```
session_register("nome_variavel") = "valor";
```

E para destruir uma sessão:

```
session_destroy();
```

PARÂMETRO	DESCRIÇÃO
<b>session_cache_expire</b>	Retorna o prazo do cache atual
<b>session_cache_limiter</b>	Obtém e/ou define o limitador do cache atual
<b>session_commit</b>	Sinônimo de session_write_close
<b>session_decode</b>	Decifra dado de sessão de uma string
<b>session_destroy</b>	Destrói todos os dados registrados em uma sessão
<b>session_encode</b>	Codifica os dados da sessão atual como uma string
<b>session_get_cookie_params</b>	Obtém os parâmetros do cookie da sessão
<b>session_id</b>	Obtém e/ou define o id de sessão atual
<b>session_is_registered</b>	Descobre se uma variável global está registrada numa sessão
<b>session_module_name</b>	Obtém e/ou define o módulo da sessão atual
<b>session_name</b>	Obtém e/ou define o nome da sessão atual
<b>session_regenerate_id</b>	Atualiza o id da sessão atual com um novo gerado
<b>session_register</b>	Registrar uma ou mais variáveis globais na sessão atual
<b>session_save_path</b>	Obtém e/ou define o save path da sessão atual
<b>session_set_cookie_params</b>	Define os parâmetros do cookie de sessão
<b>session_set_save_handler</b>	Define a sequência de funções de armazenamento
<b>session_start</b>	Inicia dados de sessão
<b>session_unregister</b>	Desregistra uma variável global da sessão atual
<b>session_unset</b>	Libera todas as variáveis de sessão
<b>session_write_close</b>	Escreve dados de sessão e termina a sessão

# Tarefas para teste de sessões em PHP

- a) Criar uma página [login.php](#) com um formulário contendo 2 inputs do tipo “text”, um de nome “user” outro de nome “pass”, um input do tipo “submit” e um input do tipo “reset”. As informações do formulário serão submetidas ao arquivo autentica.php
- b) Criar o arquivo [autentica.php](#) que fará o processo autenticação e criação de sessões.
- c) Criar o arquivo [pagina.php](#) que irá exibir informações da sessão caso esta seja criada com sucesso.
- d) Testar a relação entre essas três páginas
- e) Criar o arquivo [sessao.inc.php](#) com verificação de sessão genérica
- f) Inserir o arquivo [sessao.inc.php](#) nas páginas a fim do usuário ter acesso a elas apenas com a inserção de login e senha corretos

Semelhantes às *sessions*, porém são arquivos de texto armazenados no computador do usuário contendo um nome e valor e pode ser recuperado posteriormente pelo servidor. Quando acessamos um site pela segunda vez e o campo usuário da autenticação já está preenchido é sinal que o site se utilizou de um *cookie*.

É possível especificar um tempo de validade que quer dizer quanto tempo o cookie permanecerá na máquina do usuário.

Os navegadores possibilitam desabilitar a utilização pois estes muitas vezes são vistos como mecanismos invasivos.

Por essa razão os cookies devem ser usados com cuidado e armazenando informações realmente necessárias.



# Cookies

```
setcookie(string name [, string value [, int  
expire [, string path [, string domain [, int  
secure]]]] )
```

PARÂMETRO	DESCRIÇÃO
Name	O nome do cookie.
Value	O valor do cookie.
Expire	O tempo (em segundos) para o cookie expirar.
Path	O caminho no servidor para qual o cookie estará disponível.
Domain	O domínio que o cookie estará disponível.
Secure	Indica que o cookie deve ser transmitido sobre uma conexão segura HTTPS.

# Cookies

```
<?php
```

```
/* Este exemplo mostra a criação do cookie nome_user com  
o valor “João” que irá expirar em dois dias. Utiliza-se a  
função time() que pega o horário atual mais o número de  
segundos equivalente a quantidade de tempo que se manter  
o cookie no computador do usuário */
```

```
setcookie("nome_user","João",time()+172800);
```

```
/* Para excluir o cookie basta executá-lo novamente da  
seguinte maneira */
```

```
setcookie("nome_user");
```

```
?>
```

# Cookies

```
<?php
```

```
/* Esta função de criação de cookie deve ser  
utilizada no início da navegação sendo colocada no  
código antes das tags HTML. Estes cookies podem ser  
recuperados através da variável global $_COOKIE, mas  
não na mesma página que o criou e sim nas próximas  
solicitações da página. */
```

```
$_COOKIE["nome_user"];
```

```
?>
```

O Manual Oficial do PHP traz um capítulo com todas as funções e exemplos de sessões em PHP (CXXXV. Sessões).