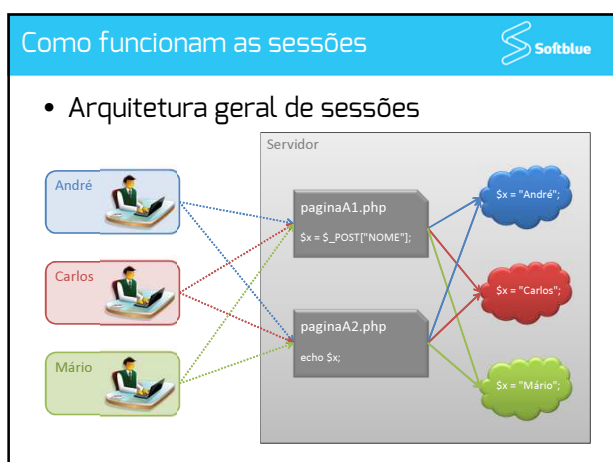




Sessões

- Sequências de acessos identificados por usuário e/ou conexão, de curto intervalo
- Permitem armazenar variáveis no servidor durante a sequência de acessos
- Vantagens
 - Acesso rápido a variáveis comuns entre os acessos da sessão
 - Identificação do usuário e/ou conexão de cada requisição (controle de acesso)
- Desvantagens
 - Uso de memória do servidor para armazenamento das sessões e variáveis
 - Duram em média poucos minutos, ou até um logout / logoff ser realizado
- Via URL ou via Cookie
 - use_trans_sid



Inicializando uma sessão



• session_start()

Sintaxe

```
(bool) session_start()
```

Importante

Ao configurar o gerenciamento de sessões do PHP por cookie, o comando **session_start()** deve ser chamado antes que qualquer dado seja enviado para a response

A partir da versão 4.3.3 do PHP, o comando **session_start()** resultará em um erro de nível **E_NOTICE** caso a sessão já tenha sido iniciada

Salvando uma variável na sessão



• Array \$_SESSION[]

Sintaxe

```
(void) $_SESSION["variável"] = valor;
```

Exemplo

```
$_SESSION["nome"] = "André";
$_SESSION["nota"] = 8;
$_SESSION["cores"] = array("branco", "azul", "cinza");

// Resultado: armazena os respectivos valores em variáveis de sessão com os nomes
// estipulados no exemplo

echo $_SESSION["nota"];
```

```
// Resultado: imprime o valor 8 na tela
```

Limpendo uma variável de sessão



• unset()

Sintaxe


```
(void) unset($_SESSION["variável"])
```

Exemplo

```
$_SESSION["nome"] = "André";
$_SESSION["nota"] = 8;
unset($_SESSION["nota"]);

// Resultado: remove a variável "nota" da sessão, mas mantém a variável "nome"
```

Excluindo uma sessão



- `session_destroy()`

Sintaxe


```
(bool) session_destroy()
```

Exemplo

```
$_SESSION["nome"] = "André";
$_SESSION["nota"] = 8;
session_destroy();

// Resultado: remove todas as variáveis da sessão
```

Cookies



- Valores e/ou variáveis armazenadas no navegador do usuário de um site durante uma sequência de acesso
- Vantagens
 - O cookie pode permanecer ativo durante minutos, horas, dias, meses ou anos
 - Não utiliza espaço em disco ou memória do servidor
- Desvantagens
 - Valores podem ser gerenciados pelos usuários com conhecimento
- Quando utilizar cookies

Como funcionam os cookies



- Arquitetura geral de cookies



Criando um cookie



- setcookie

Sintaxe
(bool) **setcookie**(nome[, valor[, prazo[, caminho[, domínio[, https[, httponly]]]]]])

Exemplo

```
setcookie("meuCookie", "Eu gosto de PHP");  
// Resultado: cria o cookie meuCookie com o valor "Eu gosto de PHP"  
  
setcookie("contalD", "6548", time() + (86400), "/sb/", ".softblue.com.br/home");  
// Resultado: cria o cookie contalD na subpasta sb com validade de 1 dia e  
// restrito para o domínio .softblue.com.br/home
```

Importante

O comando setcookie deve ser utilizado antes de qualquer saída ser enviada para a response, caso contrário não funcionará

Acessando um cookie



- Array \$_COOKIE[]

Sintaxe
\$_COOKIE(nomeDoCookie)

Exemplo

```
echo $_COOKIE["meuCookie"];  
// Resultado: imprime na tela o valor do cookie meuCookie
```

Excluindo um cookie



- setcookie com parâmetros específicos

Exemplo

```
setcookie("meuCookie", 0);  
// Resultado: deleta o cookie meuCookie do cliente remoto  
  
setcookie("contalD", 0, time() - (86400), "/sb/", ".softblue.com.br/home");  
// Resultado: deleta o cookie contalD do cliente remoto
```

Aulas práticas e manuais on-line





Assista agora as aulas práticas, que apresentam o uso dos comandos abordados nesta aula teórica.

[Clique aqui](#) para visualizar as aulas práticas disponíveis
