



**INSTITUTO FEDERAL**

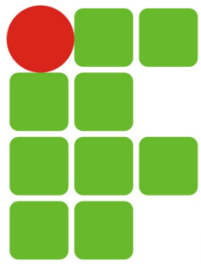
Farroupilha

Campus Avançado Uruguaiana

# Sessões

Prof. Leandro Martins Dallanora

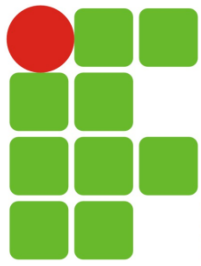
[leandro.dallanora@iffarroupilha.edu.br](mailto:leandro.dallanora@iffarroupilha.edu.br)



# Sessões - Introdução

---

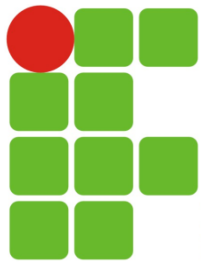
- As sessões são estruturas muito parecidas com os cookies.
- Uma das principais diferenças é de que as sessões são armazenadas no próprio servidor e não expiram a menos que o programador queira intencionalmente apagar a sessão.



# Sessões - Introdução

---

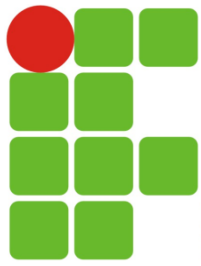
- Assim quando um usuário entra no site, podemos abrir uma sessão e nela registrar diversas variáveis, que ficarão gravadas em arquivos no servidor, onde poderão ser acessadas de qualquer página do site, enquanto a sessão estiver aberta.



# Sessões: Iniciando

---

- É necessário utilizar a função **session\_start** para iniciar qualquer sessão em PHP.
- Você deve chamar essa função antes de qualquer *output* do seu código, ou seja, antes de qualquer **echo**, **print**, **códigos HTML** e coisas do tipo.
- Geralmente, chamamos a função **session\_start** nas primeiras linhas do nosso código PHP para prevenir erros.



# Sessões: Iniciando

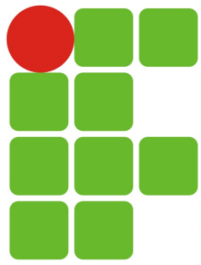
---

- Exemplo:

**// Inicia a sessão**

**session\_start();**

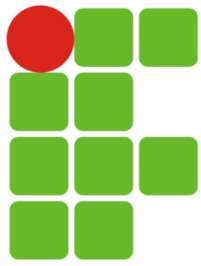
- Você sempre deve utilizar o trecho de código acima para fazer qualquer coisa relacionada a uma sessão PHP, mesmo que seja para destruí-la.



# Sessões: Definindo valores

---

- **\$\_SESSION** é um array associativo que recebe os valores que você quiser definir.
- Veja um exemplo para definir uma sessão de nome, usuário e senha. Esses dados estarão disponíveis em todas as páginas do nosso script enquanto o navegador estiver aberto ou até a sessão terminar.



# Sessões: Definindo valores

---

**// Inicia a sessão**

```
session_start();
```

**// Cria as chaves e seus valores**

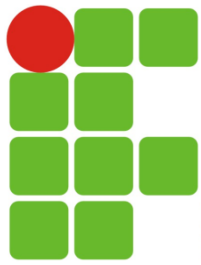
```
$_SESSION['nome'] = 'Leandro';
```

```
$_SESSION['usuario'] = 'dallanora';
```

```
$_SESSION['senha'] = '1234';
```

**// Mostra uma frase na tela**

```
echo 'Olá ' . $_SESSION['nome'];
```

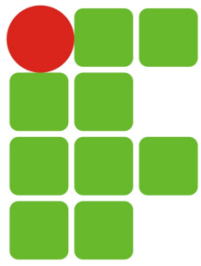


# Sessões: Definindo valores

---

- Fazendo isso, agora eu poderia criar um sistema de login para meu usuário, já que a sessão é individual para cada acesso. Isso significa que cada pessoa que acessar seu site, terá uma sessão própria, com ID e valores únicos.
- Se eu quiser acessar os dados da sessão em outros arquivos PHP dentro do meu script, basta iniciar a sessão e exibir os dados.





# Sessões: Definindo valores

---

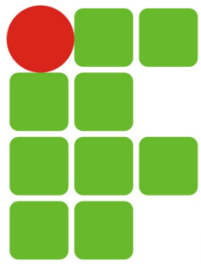
- Exemplo:

**// Inicia a sessão**

```
session_start();
```

**// Mostra uma frase na tela**

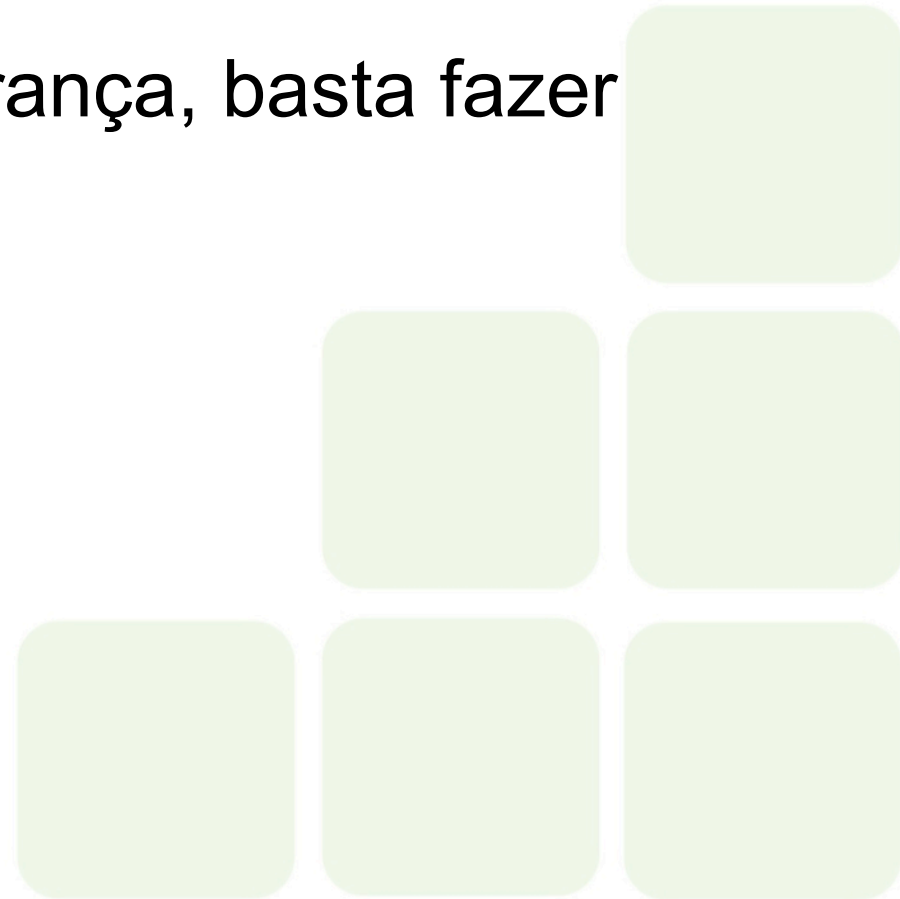
```
echo 'Olá ' . $_SESSION['nome'];
```

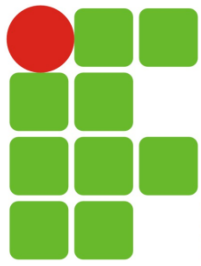


# Sessões: Validando

---

- Perceba que não estou verificando se a sessão existe.
- Se eu quiser mais segurança, basta fazer algumas verificações.





# Sessões: Validando

---

**// Inicia a sessão**

```
session_start();
```

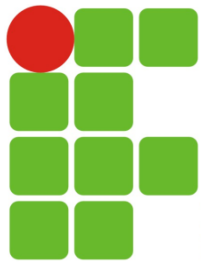
**// Verifica se a sessão do usuário não existe**

```
if ( ! isset( $_SESSION['usuario'] ) ){
```

**// Ação a ser executada: mata o script e  
manda uma mensagem**

```
exit('Usuário não está logado');
```

```
}
```

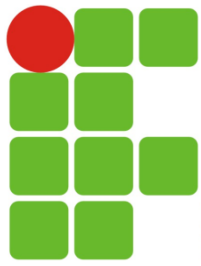


## Sessões: Validando

---

- No exemplo anterior, estamos apenas verificando se as variáveis de sessão existem.
- Poderíamos verificar ainda se elas não estão vazias, e/ou se o usuário e a senha existem na base de dados.





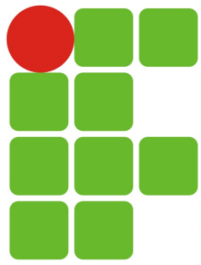
# Sessões: Finalizando

---

- Para apagar qualquer valor de uma sessão, você pode utilizar a função **unset** do PHP. Veja um exemplo:

**// Apaga a chave**

```
unset( $_SESSION['usuario'] );
```



# Sessões: Finalizando

---

- Ou podemos simplesmente finalizar a sessão inteira. Exemplo:

**// Finaliza a sessão**

**session\_destroy();**

- Destruir a sessão é sempre mais eficaz do que qualquer outro método.