



**DIO**  
CEZAR  
**GO**

# PHP Namespaces

—

# O QUE SÃO?

- possibilitam o agrupamento de classes, interfaces, funções e constantes;
- resolvem o conflito em seus nomes;
- atuam como um encapsulador para estes itens;
- seu funcionamento é equivalente ao de diretórios em sistemas operacionais;  
(dois arquivos com o mesmo nome não podem existir em um mesmo diretório, mas podem existir em diretórios diferentes)
- foi inserido a partir da versão 5.3 do PHP;

# DEFININDO NAMESPACES

---

- namespaces são declarados utilizando a palavra-chave: namespace.
- um arquivo que contenha namespace deve realizar sua declaração logo no início, antes de qualquer código;

## DECLARANDO UM NAMESPACE

```
<?php  
namespace Project;  
// ...
```

# DEFININDO NAMESPACES

---

- e ainda é possível criar uma estrutura hierárquica (como em pastas) ou seja, sub-namespaces;

## DECLARANDO UM NAMESPACE

```
<?php
namespace Project\Model;
// ...
```

# DEFININDO NAMESPACES

---

- é possível também declarar diversos *namespaces* em um arquivo;

## VÁRIOS NAMESPACES

```
<?php
    namespace Project;
    class User {}

    namespace AnotherProject;
    class User {}
?>
```

# DEFININDO NAMESPACES

---

- a seguinte sintaxe com colchetes também é válida:

## VÁRIOS NAMESPACES

```
<?php
    namespace Project {
        class User {}
    }

    namespace AnotherProject {
        class User {}
    }
?>
```

# UTILIZANDO NAMESPACES

---

- nome qualificado → `$obj = new User();`
- tenta achar a classe no namespace atual;

## NOME NÃO QUALIFICADO

```
<?php
    namespace Project\Model;
    class User {}
    $var = new User();
?>
```

# UTILIZANDO NAMESPACES

---

- nome não qualificado → `$obj = new User();`
- tenta achar a classe no namespace atual;
- caso o namespace atual seja `Project\Model` tenta achar `Project\Model\User`;

## NOME NÃO QUALIFICADO

```
<?php
    namespace Project\Model;
    class User {}
    $var = new User;
?>
```



# UTILIZANDO NAMESPACES

---

- nome qualificado → `$obj = new Model\User();`
- caso o namespace atual seja Project tenta achar `Project\Model\User`

## NOME NÃO QUALIFICADO

```
<?php
    namespace Project;
    $var = new Model\User;
?>
```

# UTILIZANDO NAMESPACES

---

- nome absoluto → `$obj = new \Project\Model\User();`
- solicita o nome absoluto `Project\Model\User`

## NOME NÃO QUALIFICADO

```
<?php
    namespace Project;
    $var = new \Project\Model\User;
?>
```

# APELIDOS

---

- uma característica importante ao se trabalhar com *namespaces* é a possibilidade de atribuição de apelidos;
- a palavra chave use é utilizada para importar classes, interfaces ou namespaces através de seus nomes;
- para adicionar apelidos utilizamos a palavra-chave *as*, escolhendo um nome mais acessível;

# APELIDOS

---

- é possível realizar múltiplas importações na mesma linha, separadas por vírgula;

## APELIDOS

```
<?php
    use Project\Model\User, Project\Model\Post as Article;
    $var = new Article();
    // Resultado: Project\Model\Post
    $user = new User();
    // Resultado Project\Model\User
?>
```

# **MATERIAIS COMPLEMENTARES**

---

- <https://goo.gl/jNDEd4>