

DAWS – 2º

Namespaces

JESÚS MOLINA HERNÁNDEZ

jmolina@florida-uni.es

Índice

1. **Caso práctico**
2. **¿Qué es un namespace?**
3. **Beneficios de los namespaces en PHP**
4. **Sintaxis en PHP**
5. **Reglas básicas para usar namespaces**
6. **Cómo acceder a una clase en un namespace**
7. **Ejemplos de namespaces en PHP**

Caso práctico

Varios compañeros vais a trabajar en una aplicación y os repartís el trabajo. Y resulta que se os ocurre llamar a algunas clases que hacen cosas distintas del mismo modo. Cuando juntéis el código, eso sería un desastre.

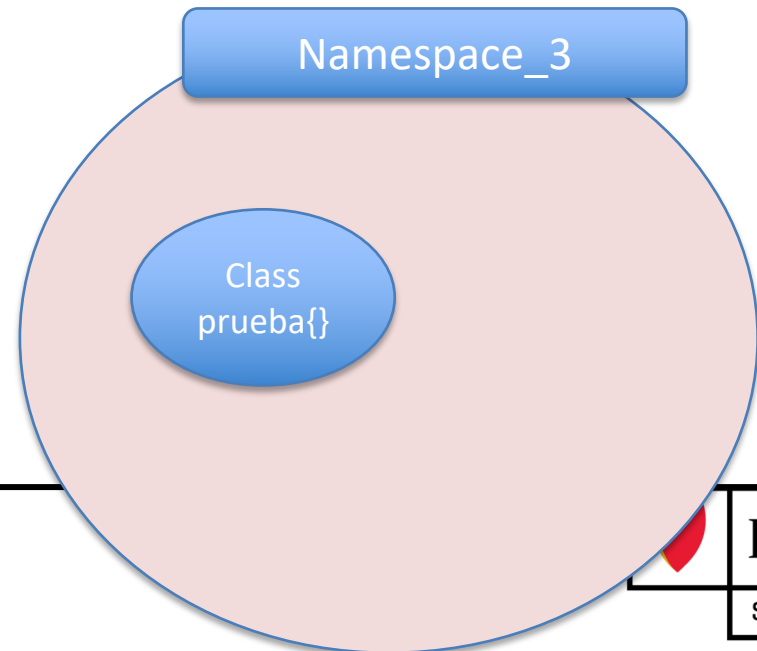
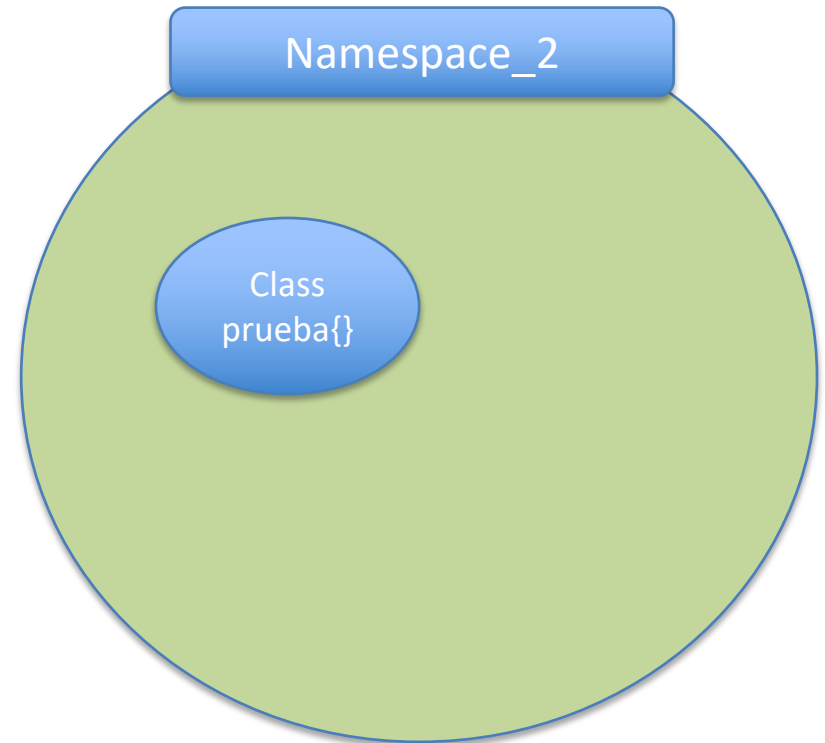
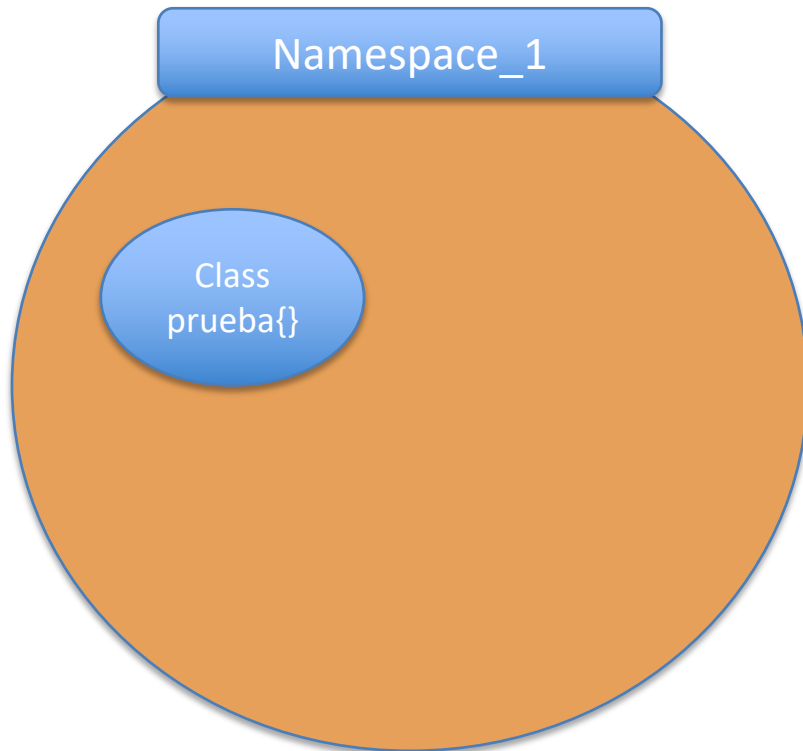
Entre unos cuantos colegas parece raro, pero imaginad cuando incluís código de terceros, librerías que se usan a menudo. Ahí las probabilidades aumentan.

Conflicto de nombres repetidos...

Solución: Namespaces

Namespaces

¿Qué es un namespace?



Namespaces

Beneficios de los namespaces en PHP

Evitar conflictos de nombres: En aplicaciones grandes o al integrar múltiples bibliotecas, puede haber clases o funciones con el mismo nombre. Los namespaces ayudan a diferenciarlos.

Organización del código: Permiten agrupar clases, funciones y constantes relacionadas en un mismo espacio lógico, mejorando la estructura del código.

Facilitar la reutilización del código: Al usar namespaces, puedes reutilizar librerías de terceros sin preocuparte por conflictos de nombres.

Se usan mucho en
frameworks (laravel,
symphony)

Sintaxis en PHP

Los **namespaces** se declaran al inicio del archivo PHP con la palabra clave **namespace**.

```
namespace NombreDelNamespace;  
  
class MiClase {  
    // Código de la clase  
}
```

Reglas básicas para usar namespaces

Declaración única:

- El **namespace** debe ser declarado al principio del archivo. Solo puede haber **un namespace** principal por archivo.

Acceso a las clases y funciones:

- Para usar una clase o función en un **namespace** diferente, se puede:
 - Utilizar el nombre **completo** del namespace.
 - Usar la declaración **use** para importar el namespace y simplificar el acceso.

Separador de espacios:

- El separador de los elementos de un namespace es la barra invertida \.

Cómo acceder a una clase en un namespace

Dos opciones

(1) Nombre completo

```
$obj = new \MiNamespace\MiClase();
```

(2) 'use' y 'use ... as'

```
use MiNamespace\MiClase;  
  
$obj = new MiClase();
```

```
use MiApp\Utils\Validator as CustomValidator;  
use Vendor\Validation\Validator as VendorValidator;
```

Con Alias

Buen uso de namespaces en proyectos grandes

Supongamos que estás creando una librería llamada MiLibreria y tienes diferentes módulos dentro de esta librería, como Utilidades y Servicios. Aquí te muestro cómo organizar la estructura de carpetas y namespaces:

Estructura de carpetas:

```
bash
/mi_libreria
├── /src
│   ├── /Utilidades
│   │   └── Funciones.php
│   └── /Servicios
│       └── ServicioDeArchivo.php
└── index.php
```

Definir el namespace principal y los subnamespaces

- **Namespace principal: MiLibreria**
- **Subnamespaces:** Dentro del directorio src, cada carpeta tendrá un subnamespace:
MiLibreria\Utilidades
MiLibreria\Servicios

```
namespace MiLibreria\Utilidades;

class Funciones {
    public static function saludar($nombre) {
        return "Hola, " . $nombre;
    }
}
```

```
namespace MiLibreria\Servicios;

class ServicioDeArchivo {
    public function guardarArchivo($nombreArchivo, $contenido) {
        file_put_contents($nombreArchivo, $contenido);
        echo "Archivo guardado: " . $nombreArchivo;
    }
}
```

Namespaces

Ejemplos de namespaces en PHP

Ver ejemplos en PHPStorm

- *Sencillo*
- *Avanzado*