Espacios de nombres en PHP (y III). Jerarquía de namespaces.

En el primer artículo de esta serie explicábamos que los espacios de nombres son una forma de clasificar elementos equivalente, en cuanto al concepto, a lo que son los directorios para los ficheros. De este modo podemos tener elementos, como ya hemos aprendido, con el mismo nombre, en diferentes espacios de nombres. A la hora de referenciar un elemento, lo hacemos precediéndole, en su caso, del espacio de nombres al que dicho elemento pertenece.
Al igual que ocurre con los directorios, podemos especificar una ruta de acceso de tres modos:

Sin especificar un espacio de nombres. En ese caso se entiende que el elemento referenciado se encuentra en el espacio de nombres actual. Es lo mismo que no usar espacio de nombres. Esto se conoce como **nombre sin cualificar**. Es el equivalente, cunado hablamos de ficheros y directorios a llamar a un fichero sólo por su nombre, asumiendo que está en el mismo directorio en el que estamos trabajando.
Especificando un espacio de nombres, la barra invertida, y el elemento al que queremos referirnos. Esto se conoce como **nombre cualificado**. El espacio de nombres que usamos "depende" del espacio de nombres actual. Es el equivalente a usar una ruta relativa para referirnos a un fichero, cuando está en un directorio diferente del actual.
Especificando un espacio de nombres precedido de una barra invertida, y separado del elemento con una barra invertida. Esto se conoce como **nombre completamente cualificado**. Es el equivalente a usar rutas absolutas cuando buscamos un fichero en un árbol de directorios.

NOMBRES SIN CUALIFICAR
Veamos el siguiente código, llamado nombres_sin_cualificar.php:

```
<?php
 namespace sin_cualificar;

 const CONSTANTE = 'Valor de constante referida como nombre sin cualificar.'.'<br>';

 function funcionDePrueba(){
  echo 'Esta es una función referida como nombre sin cualificar.'.'<br>';
 }

 class ClaseDePrueba{
  public static function metodoDeClase(){
   echo 'Este es un método de clase referido como nombre sin cualificar.'.'<br>';
  }
 }

 echo CONSTANTE;
 funcionDePrueba();
 ClaseDePrueba::metodoDeClase();
?>
```

Como ves, todos los elementos del script se hallan bajo el espacio de nombres que hemos llamado sin_cualificar (no puede haber

espacios en blanco en los nombres de namespaces). En las tres última líneas (resaltadas en el listado) referenciamos los elementos. Como lo hacemos desde el mismo espacio de nombres, no es preciso indicar dicho espacio. Son nombres sin cualificar que, como decíamos al principio, equivalen, en nuestro ejemplo comparativo con archivos y directorios (salvando las diferencias) a las rutas implícitas.

NOMBRES CUALIFICADOS

Estos son los que hemos estado usando hasta ahora. Se llama a un elemento con un namespace relativo (esto es, directamente accesible) desde el contexto actual. Podemos ver un ejemplo es nombres_cualificados.php:

```
<?php
 include 's1.php';

 const CONSTANTE = 'Valor de constante no perteneciente a un espacio de nombres.'.'<br>';
 function funcionDePrueba(){
  echo 'Esta es una función no perteneciente a un espacio de nombres.'.'<br>';
 }
 class ClaseDePrueba{
  public static function metodoDeClase(){
   echo 'Este es un método de clase no perteneciente a un espacio de nombres.'.'<br>';
  }
 }

 echo CONSTANTE;
 funcionDePrueba();
 ClaseDePrueba::metodoDeClase();

 echo ns1CONSTANTE;
 ns1funcionDePrueba();
 ns1ClaseDePrueba::metodoDeClase();
?>
```

Observa en el primer grupo de líneas resaltadas como nos referimos a los elementos que son declarados en el propio contexto en el que estamos trabajando. En el segundo grupo (las tres última líneas) nos referimos a los elementos que pertenecen al espacio de nombres ns1 y que se han obtenido al incluir, en este script, el que hemos llamado s1.php, cuyo listado es el mismo que el de los anteriores artículos, por lo que no lo reproducimos aquí. Como puedes ver, el espacio de nombres ns1 se referencia desde el contexto actual, por lo que es el equivalente, como ya hemos comentado, a usar una ruta relativa.

NOMBRES COMPLETAMENTE CUALIFICADOS

Usamos esta nomenclatura cuando necesitamos referirnos a un elemento de un espacio de nombres expresando toda la ruta de espacios de nombres entre el contexto actual y el elemento al que nos vamos a referir, como puedes ver en el script nombres_completamente_cualificados.php:

```
<?php
 namespace principal;
 include 's1.php';

 const CONSTANTE = 'Valor de constante no perteneciente a un espacio de nombres.'.'<br>';
 function funcionDePrueba(){
  echo 'Esta es una función no perteneciente a un espacio de nombres.'.'<br>';
 }
 class ClaseDePrueba{
  public static function metodoDeClase(){
   echo 'Este es un método de clase no perteneciente a un espacio de nombres.'.'<br>';
  }
 }
```

```
 echo CONSTANTE;
 funcionDePrueba();
 ClaseDePrueba::metodoDeClase();

 echo ns1CONSTANTE;
 ns1funcionDePrueba();
 ns1ClaseDePrueba::metodoDeClase();
?>
```

Observa las líneas resaltadas. El espacio de nombres ns1 no es directamente accesible, ya que las llamadas a sus elementos se hallan, a su vez, dentro de contexto de otro espacio de nombres, que hemos llamado principal. Por esta razón, no podemos referirnos a ellos sino con un nombre completamente cualificado, lo que sería el equivalente a una ruta absoluta.
Los que estéis familiarizados con el uso de Linux entenderéis con más facilidad esta diferencia sutil, ya que el árbol de directorios de este SO se gestiona, en muchos casos, con rutas absolutas que usan la misma notación que los nombres completamente cualificados de los espacios de nombres.

JERARQUÍA DE ESPACIOS DE NOMBRES

El empleo de espacios de nombres con nombres completamente cualificados nos permite establecer una jerarquía de anidamiento de namespaces. Esto es especialmente útil cuando tenemos una gran cantidad de elementos que queremos organizar en distintos espacios de nombres, a fin de tener un código limpio y bien organizado. Observa el script jerarquia.php:

```
<?php
 namespace principal;
 include 's2.php';

 const CONSTANTE = 'Valor de constante no perteneciente a un espacio de nombres.'.'<br>';
 function funcionDePrueba(){
  echo 'Esta es una función no perteneciente a un espacio de nombres.'.'<br>';
 }
 class ClaseDePrueba{
  public static function metodoDeClase(){
   echo 'Este es un método de clase no perteneciente a un espacio de nombres.'.'<br>';
  }
 }

 echo CONSTANTE;
 funcionDePrueba();
 ClaseDePrueba::metodoDeClase();

 echo principalns2CONSTANTE;
 principalns2funcionDePrueba();
 principalns2ClaseDePrueba::metodoDeClase();
?>
```

Como ves, se incluye el script s2.php, cuyo listado es el siguiente:

```
<?php
 namespace principalns2;

 const CONSTANTE = 'Valor de constante en ns2'.'<br>';
 function funcionDePrueba(){
  echo "Esta en la función del script en ns2".'<br>';
 }
 class ClaseDePrueba{
```

```
  public static function metodoDeClase(){
   echo "Esta es una función de clase en ns2".'<br>';
  }
 }
?>
```

Observa la línea resaltada. Este script define el espacio de nombres ns2, pero le estamos indicando que éste es, a su vez, parte del espacio de nombres principal. En el script jerarquía.php tenemos, por lo tanto, que emplear nombres completamente cualificados para referirnos a los elementos de ns2.

Te dejo todos los scripts en este enlace, para que puedas descargarlos y experimentar con ellos, a fin de familiarizarte con los conceptos expuestos en este artículo.