

1 Reglas básicas para trabajar con clases en PHP5

Reglas básicas para trabajar con clases bajo PHP5

- Una clase debe estar en un solo archivo/fichero, y el nombre del archivo/fichero debe ser el mismo que el nombre de la clase. Por ej, "Cliente.php" y la clase internamente debe llamarse "Cliente", en "ningún caso" los nombres deben diferir.
- Cada clase debe manejar su relación con otras clases de forma independiente de las clases con las cuales interactúa: para que una clase sea consistente esta debe saber a qué clases llamar para poder trabajar, sin importar que otra clase con la que debe relacionarse comparta una misma relación. A pesar que PHP permita colocar todos los include/require a las clases en un archivo centralizado (como muchas veces se hace en "index.php") , debemos evitarlo. Lo único que logramos es que la clase sea inconsistente en otro contexto al fallar porque ella no sabe resolver sus relaciones (imaginemos que una de esas clases no se usará en index.php, por lo tanto no estará la lista de include/require de clases que necesita para poder funcionar).
- En la relación entre las clases debes usar **require_once**, lo que resuelve si dos clases que se relacionan entre sí dependen a su vez de la misma clase, por consiguiente solo se invoca una vez, y seguimos manteniendo la independencia de la clase, ya que si nos llevamos una de esas clases a otro contexto, esta seguirá funcionando correctamente y permitirá el reuso normal de la misma sin dar ningún tipo de error de "clase ya requerida" (si usas solo include o require, no puedes llamar dos veces a la misma clase, por eso existen include_once y require_once).
- En un mismo contexto, nunca redefinir la misma clase, ya que pierde consistencia el diseño del sistema. Por ej, en un sistema contable tener una clase Cliente que según la clase que la llame puedan "reescribir" internamente esa clase para que su diseño sea distinto a la clase original. Una clase debe ser única en un mismo sistema, no puede ser que existan dos versiones de la implementación de una misma clase.

Estándares a la hora de nombrar variables:

- Si la variable tiene visibilidad privada se le suele añadir un `_` delante, por ejemplo:

```
<?php
class Match{
// La variable $_dni es privada.
private $_dni;
public $nombre;
....

}
?>
```

ENLACES MUY RECOMENDABLES:

- [Información sobre estándares.](#)
- [Why you are a bad PHP programmer.](#)