

Organización del proyecto del lenguaje PHP

Include vs Require

- Las páginas PHP pueden llegar a ser muy extensas. Es conveniente estructurar el código de una página en **distintos ficheros**.
- Podemos utilizar las funciones **include** y/o **require** para vincular unos ficheros con otros.
 - Si utilizamos **require** y el fichero que estamos incluyendo no existe o provoca un error la ejecución del código se para, ya que provoca un **error fatal**.
 - Si utilizamos **include** en lugar de producir un error fatal producirá un **warning**, por lo que el resto del código será ejecutado.
- Usaremos **require** cuando la **ejecución del código que incluimos es indispensable** para el correcto funcionamiento de la página, **en caso contrario** usaremos **include**.

Separar lógica de presentación (vistas)

- Vamos a tratar de separar la parte del código que se encarga de obtener y validar datos, así como cualquier otro procesamiento que haya que realizar con los mismos (**lógica de negocio**), de la parte que se encarga de mostrar esos datos (**vistas**).
- **La vista se vinculará** a los documentos php **mediante un require**.
- En la vista podremos acceder a los datos que se han obtenido en la parte de lógica de negocio.

Mostrar arrays en las Vistas

- Es muy habitual tener arrays de datos que tenemos que mostrar en la vista.
- Para que las vistas sean más intuitivas es conveniente utilizar estructuras foreach de la siguiente forma:

```
<?php foreach ($persons as $person) : ?>  
    <li><?= $person ?></li>  
<?php endforeach; ?>
```

Mostrar arrays asociativos en las Vistas

- En ocasiones los arrays a visualizar son arrays asociativos
- Los recorreremos de la siguiente forma:

```
<?php foreach ($persons as $key=>$value) : ?>  
    <li><strong><?= $key; ?></strong><?=$value; ?></li>  
<?php endforeach; ?>
```

Condiciones sencillas en las vistas

- Es importante que las vistas no incorporen lógica de negocio.
- En ocasiones necesitamos introducir condiciones sencillas:

```
<li>  
  <strong>  
    <?= $person['married'] ? 'Casado' : 'Soltero'; ?>  
  </strong>  
</li>
```

Condiciones en las vistas

- Cuando el html a generar en las condiciones es más complejo:

```
<?php if ($person['married']) : ?>  
    <span class="icon">#10004;</span>  
<?php else : ?>  
    <span class="icon">#10008;</span>  
<?php endif; ?>
```

Partials

- Normalmente todas las páginas de una aplicación web tienen una **parte común** (cabecera, pie, barra de navegación, etc.).
- Estas partes comunes las sacaremos a un fichero a parte (**partials**).
- Es conveniente meterlos dentro de una carpeta “partials” por organización.
- Después **incluiremos (include) los partials en las páginas principales.**

Separar el código reutilizable

- Las funciones reutilizables las llevaremos a ficheros independientes (normalmente dentro de un directorio “utils”).
- Ejemplo: función de utilidad para mostrar el contenido de un elemento:

```
function dd($data)
{
    echo '<pre>';
    die(var_dump($data));
    echo '</pre>';
}
```