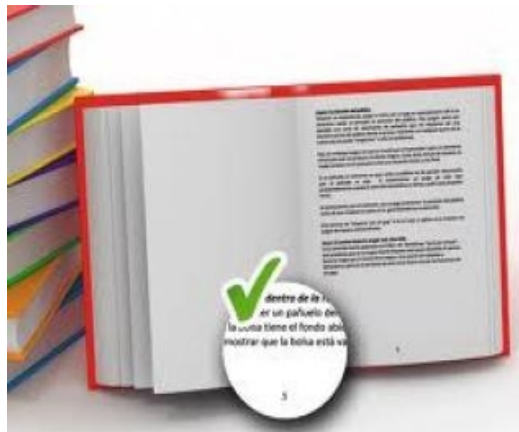


## PAGINACIÓN

### Vamos a dividir el resultado de nuestras consultas en distintas páginas.

En caso de que el número de registros que debemos mostrar en una página web sea demasiado grande, no es conveniente presentar al usuario tanta información a la vez. Primero, porque es incómodo para él tener que ir desplazándose por la página y segundo, porque esto implicará que la página tarde bastante tiempo al tener que generarse toda la tabla donde mostramos los datos. Normalmente es preferible dividir el resultado en varias páginas, mostrando en cada una de ellas un número más reducido de registros. Esto se conoce como **paginación**. Podemos proporcionar paginación



fácilmente utilizando el modificador **LIMIT** de una consulta **SELECT**. **LIMIT** establece un límite en cuanto al número de registros que devolverá **SELECT**.

Su sintaxis es **LIMIT [desplazamiento,] numerodefilas**. En *desplazamiento* podemos indicar desde qué registro queremos obtener los resultados y en *numerodefilas*, el número de filas que devolverá la consulta. Por ejemplo, **LIMIT 0 5** devolverá las 5 primeras filas del **SELECT**, mientras que **LIMIT 10, 5** devolverá 5 filas a partir de la número 10 del resultado.

**Por lo tanto, para hacer paginación debemos conocer:**

- 1 El número total de filas para saber si es necesaria la paginación.
- 2 El número de filas/registros que deseamos mostrar por página.
- 3 El desplazamiento.

Calcularemos el número total de filas que devuelve la consulta (ya sabéis cómo), es decir, el número total de productos a mostrar, y lo guardamos, por ejemplo, en la variable **\$total\_articulos**.

Ahora necesitamos decidir el número de filas que se mostrará en cada página. En este ejemplo, vamos a mostrar 5 productos por página; **\$num\_filas = 5**.

El desplazamiento va variando conforme avancemos o retrocedamos por las páginas:

```
if (isset($_GET["desplazamiento"]))
    $desplazamiento = $_GET["desplazamiento"];
else $desplazamiento = 0;
///Vale, esto de arriba es un poco chapucero para nosotros, ahora que ya conocemos
//el operador Fusion Null:
$desplazamiento = $_GET["desplazamiento"] ?? 0;
//muho mejor así, verdad?
```

```
$resultado = mysqli_query($con, "SELECT * FROM productos
                                ORDER BY $orden LIMIT $desplazamiento, num_filas");
/// ... o con PDO
//$orden será el campo por el que queramos ordenar
```

Cómo veis, el desplazamiento será 0 o el indicado mediante la URL de la página en el campo *desplazamiento*. Conociendo este detalle, ya hemos podido completar la consulta SELECT indicando el modificador LIMIT.

Sólo nos quedaría ver la forma que proporcionaremos al usuario para que pase de una página a otra en el listado de los productos. Normalmente, esto se consigue mediante un par de hipervínculos situados en la parte inferior y/o superior de la tabla: uno para volver a la página anterior y otro para la siguiente (sólo en el caso en que existan registros antes o después, claro):

```
if ($desplazamiento > 0) {
    $prev = $desplazamiento - $num_filas;
    $url = $_SERVER["PHP_SELF"] . "?desplazamiento=$prev";
    echo "<A HREF='$url'>Página anterior</A>";
}
if ($total_articulos > ($desplazamiento + $num_filas)) {
    $prox = $desplazamiento + $num_filas;
    $url = $_SERVER["PHP_SELF"] . "?desplazamiento=$prox";
    echo "<A HREF='$url'>Próxima página</A>";
}
```

En el caso del **hipervínculo a la página anterior**, comprobamos si el desplazamiento es mayor de 0, lo que quiere decir que no es la primera página; por lo que tiene sentido este hipervínculo, y construimos el código HTML correspondiente. Fijaos que el hipervínculo apuntará a la misma página pero pasando la información necesaria de desplazamiento.

Lo mismo para el **hipervínculo a la próxima página**. La cuestión es calcular correctamente el desplazamiento en función de la página en la que nos encontremos, es decir, del desplazamiento actual (que la primera vez será 0).

Sin embargo, si diéramos la posibilidad al usuario de ordenar el resultado por algún campo en concreto, tal como tenemos el código de la página en estos momentos no es correcto, porque cada vez que se pulsa en uno de esos hipervínculos, se pierde el orden en que se mostraban los productos. Existen varias formas de solucionar esto, pero una manera sencilla sería indicar también el criterio de ordenación que se está siguiendo en los hipervínculos que nos dan acceso a la página anterior y siguiente.

```
$url = $_SERVER["PHP_SELF"] . "?orden=$orden&desplazamiento=$prev";
$url = $_SERVER["PHP_SELF"] . "?orden=$orden&desplazamiento=$prox";
```

Bastante sencillo y práctico, verdad?