

Codificación completa y correcta utf-8

Para controlar el problema de la codificación web y con ello evitar la aparición de símbolos raros correspondientes a tildes, la letra ñ o caracteres especiales hay que tener en cuenta varias causas:

- Codificación por defecto con la que arranca apache.
- Hay que indicar en php.ini para que php trabaje con utf-8.
- Codificación utf-8 del código html.
- Codificación con la que se crean y guardan los archivos.
- Utilizar las funciones de tratamiento de strings con mb.
- Codificación el SGBD, en concreto mysql.

No mezclar tipos de codificación, uno de los problemas habituales es la mezcla de codificación utf-8 e iso. Lo más recomendable es usar utf-8, por lo tanto hacerlo todo en utf-8.

Configurar la codificación del servidor apache con utf8

Afecta a las cabeceras que envía el servidor donde informa del tipo de codificación empleado.

Podemos hacerlo de dos maneras.

- Añadiendo la directiva AddDefaultCharset en el fichero httpd.conf de Apache. Esto lo haremos editando el citado fichero y añadiendo al final del mismo a línea AddDefaultCharset utf-8.
- Añadiendo la directiva anterior en el fichero htaccess.

En el caso de que no tengamos acceso al fichero de configuración del servidor podremos añadir esta directiva en el fichero htaccess del directorio raíz de nuestra web.

Configurar la codificación de php en utf8

PHP siempre devuelve una codificación de caracteres por defecto, la cual está definida con la cabecera Content-type.

Siempre hay que especificar del lado servidor el encabezado y la codificación. Para ello hay que utilizar la función de header() antes de generar cualquier tipo de contenido, por ejemplo:

```
header ('Content-type: text/html; charset=utf-8');
```

Lo más común es forzar el charset a utf-8 con la función header() desde nuestro código, pero siempre se puede especificar la codificación de la cabecera editando la configuración del php.ini donde debemos asegurarnos que la siguiente línea no está comentada.

```
default_charset = "utf-8"
```

Configurar la codificación html en utf8

Siempre se debe especificar la codificación utilizada para una página HTML o XML. De lo contrario, nos arriesgamos a que los caracteres del contenido aparezcan mal. Esto sería un problema para los usuarios de la web, y también las máquinas, los crawlers y robots que analizan el contenido de nuestra web.

DWES 2019-20

Codificación UTF-8

Dentro de la etiqueta **head** añadiremos la etiqueta **meta** correspondiente según la versión de html utilizada.

Crear y guardar los archivos en utf8

También hay que tener en cuenta como se guardan los ficheros que se van creando.

No olvidar que los ficheros van a ser ejecutados por php bajo un servidor apache. Si se trabaja en windows el IDE usará la codificación del sistema operativo por defecto, iso. Por lo general esto es algo que se puede configurar en cualquier IDE o editor de código. En nuestro caso nosotros configuraremos Eclipse para que almacene todos los ficheros del workspace en utf-8.

- Configurar el workspace de Eclipse para que los ficheros incluidos en el estén codificados en

UTF-8. Con esto los ficheros que tengamos dentro de este workspace estarán codificados en UTF- 8.

Para ello Windows → Preferencias → General → Workspace → Text File Encoding

Utilizar funciones de tratamiento de cadenas multibyte_mb

Una vez hechos los cambios anteriores comenzaremos a utilizarlas funciones multibyte para tratamiento de cadenas.

Por ejemplo el resultado de ejecutar el ejemplo siguiente sería 7 en el caso de strlen y 5 en el caso de mb_strlen.

```
$frase="cañón"; echo strlen($frase);  
echo mb_strlen ($frase);
```

Configurando el servidor mysql con utf8

Dentro de la codificación de mysql hay otros puntos que hay que seguir para arreglar los problemas que haya con la codificación. Ya que mysql cuando conecta lo hace con un tipo de codificación, cuando lee lo hace con un tipo de codificación y cuando guarda lo hace con otro tipo de codificación. Todos tienen que estar en utf-8.