

# Creación de un archivo de texto.

Una actividad fundamental es poder registrar información en el servidor (no como hemos estado haciendo hasta el momento generando sólo una página con los datos cargados).

Para la registración de datos en el servidor disponemos de dos herramientas que se complementan en muchos casos (archivos de texto y bases de datos).

En este apartado veremos como crear un archivo de texto y añadir datos al mismo.

Lo presentaremos al tema resolviendo un problema: Implementación de un libro de visitas.

Para resolver este problema plantearemos dos páginas, un formulario para realizar la carga del nombre del visitante y sus comentarios (disponemos un objeto de tipo "text" y otro de tipo "textarea"):

```
<html>
<head>
<title>Problema</title>
</head>
<body> <form action="pagina2.php" method="post">
Ingrese su nombre:
<input type="text" name="nombre">
<br>
Comentarios:
<br>
<textarea name="comentarios" rows="10" cols="40">
</textarea>
<br>
<input type="submit" value="Registrar">
</form>
</body>
</html>
```

Este formulario es similar a los planteados en problemas anteriores, sólo le hemos agregado al control textarea, las propiedades rows y cols que dimensionan el mismo en la pantalla:

```
<textarea name="comentarios" rows="10" cols="40">
</textarea>
```

Veamos ahora la página (pagina2.php) que graba los datos cargados en el formulario en un archivo:

```
<html>
<head>
<title>Problema</title>
</head>
<body>
<?php
    $ar=fopen("datos.txt","a") or
        die("Problemas en la creacion");
    fputs($ar,$_REQUEST['nombre']);
    fputs($ar,"\n");
    fputs($ar,$_REQUEST['comentarios']);
    fputs($ar,"\n");
```

```

    fputs($ar, "-----
");
    fputs($ar, "\n");
    fclose($ar);
    echo "Los datos se cargaron correctamente.";
?>
</body>
</html>

```

En primer lugar creamos o abrimos el archivo de texto "datos.txt". El segundo parámetro de la función fopen indica la forma de apertura de archivo "a" (lo crea o si ya existe el archivo lo abre para añadir datos al final), "w" (crea el archivo de texto, si existe borra su contenido) y la última forma de apertura del archivo es "r" (abre el archivo para su lectura).

Como en este problema nos interesa que el archivo vaya creciendo con los datos que aportan los visitantes al sitio lo abrimos para añadir, parámetro "a".

La función retorna una referencia al archivo, la almacenamos en una variable.

Si el archivo no se puede abrir, se ejecuta la instrucción que se encuentra luego del operador "or" en nuestro caso llamamos a la función die que finaliza la ejecución del programita PHP mostrando como mensaje el texto que le pasamos a dicha función.

```

$ar=fopen("datos.txt","a") or
    die("Problemas en la creacion");

```

Para la grabación de datos utilizamos la función fputs que tiene dos parámetros: la referencia al archivo donde grabamos y el string a grabar.

```

fputs($ar,$_REQUEST['nombre']);
fputs($ar, "\n");

```

Para el salto de línea en el archivo de texto, usamos los caracteres \n. De esta forma cuando leamos el archivo de texto lo haremos línea a línea. Cuando dejamos de trabajar con el archivo llamamos a la función fclose.

Hay que tener muy presente que el archivo se almacena en el servidor y no en la máquina de la persona que está navegando. Es decir, no vaya al explorador de archivos para ver donde se almacenó "datos.txt", tenga en cuenta que está en la máquina donde se ejecutó el script de PHP. Luego veremos como leer el contenido del archivo y mostrarlo en otra página del sitio.

## Lectura de un archivo de texto.

Para la lectura de un archivo de texto contamos con la función fgets. Además debemos habrir el archivo para lectura.

Para mostrar por pantalla el contenido del archivo "datos.txt" creado en el punto anterior tenemos el siguiente programa:

```

<html>
<head>
<title>Problema</title>
</head>

```

```

<body>
<?php
    $ar=fopen("datos.txt","r") or
        die("No se pudo abrir el archivo");
    while (!feof($ar))
    {
        $linea=fgets($ar);
        $lineasalto=nl2br($linea);
        echo $lineasalto;
    }
    fclose($ar);
?>
</body>
</html>

```

Lo primero que debemos identificar es la forma de apertura del archivo:

```

$ar=fopen("datos.txt","r") or
    die("No se pudo abrir el archivo");

```

El segundo parámetro de fopen es "r" es decir read (apertura para lectura), si el archivo no existe por ejemplo se ejecuta la función die que finaliza el programa mostrando el string correspondiente.

La función feof retorna true si se ha llegado al final del archivo en caso contrario retorna false. Para que se impriman todas las líneas del archivo se plantea una estructura repetitiva que se ejecuta mientras no se llegue al final de archivo:

```

while (!feof($ar))

```

Dentro de la estructura repetitiva leemos una línea completa del archivo de texto con la función fgets:

```

    $linea=fgets($ar);

```

La variable \$linea contiene una línea completa del archivo de texto, inclusive el salto de línea (\n)

Como el navegador no hace un salto de línea con este caracter, debemos convertir dicho caracter a la marca <br> propia de HTML. La función que realiza esta actividad se llama nl2br (new line to br)

El resultado se almacena en una nueva variable que es la que realmente imprimimos:

```

    $lineasalto=nl2br($linea);
    echo $lineasalto;

```