

TEMA 3

FORMULARIOS. MANEJO DE FICHEROS. DE VARIABLES. SERVIDOR

E6_LEEFICH

EJERCICIOS

Lectura de Archivos

Lectura Completa

1. Crear un fichero *E6_lee1.php* en el directorio de trabajo que le envíe al navegador el contenido del fichero *E6_mensajes1.txt*. Utilizar para ello la función *readfile()*.

Previamente abrir un editor como el bloc de notas y generar un archivo de texto llamado *E6_mensajes1.txt* que contenga informaciones con un formato similar al siguiente:

```
Hola mundo!!<br>Primera línea del saludo.<br>Segunda línea del saludo.<br>
```

El fichero *E6_mensajes1.txt* se encuentra en el directorio actual del proyecto.

Nuestro archivo *E6_lee1.php* se encargará de leer la información del archivo *E6_mensajes1.txt* usando *readfile()*

Nota: cuando se trabaja con *readfile()* no hay que abrir y cerrar el fichero. Ver manuales php.

Salida. Muestro el contenido del archivo de texto creado inicialmente con el editor

utiliza readfile()

El contenido del fichero mensajes.txt es

Nota: vemos como readfile() no requiere apertura y cierre del fichero.

Hola mundo!!

Primera línea del saludo.

Segunda línea del saludo.

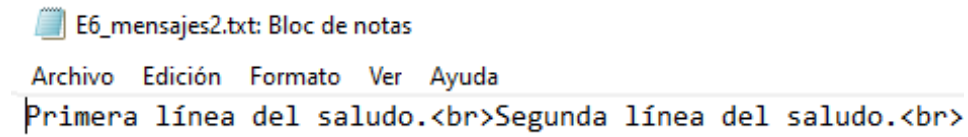
Código Plantilla:

```
// Definir variable cadena con nombre y ruta del archivo
// Controlar si el fichero existe con file_exists
// En caso de que el fichero exista, mostrar su contenido con readfile
// No hace falta cerrar archivo
.....
```

Lectura Binaria (indicando número de bytes)

2. Crear archivo E6_lee2.php

Crear desde el bloque de notas un archivo que contenga varias líneas de texto separadas con



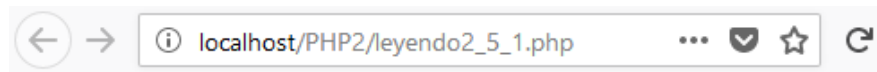
Leer el contenido del fichero *E6_mensajes2.txt*:

- Con `fread()`.
- Utilizaremos `filesize()` para leer la totalidad del archivo con una sola lectura

Añadiremos el operador @ a la operación `fopen()`. Ver manuales php.

<https://www.php.net/manual/es/language.operators.errorcontrol.php>

Nota: cuando se trabaja con `fread()`, sí que es necesario abrir y cerrar el fichero. Consulta manuales php.



utiliza `fread()`

Nota: vemos como `fread()` sí que requiere apertura y cierre del fichero.

El contenido del fichero *mensajes2_5_1.txt* es

Primera línea del saludo.

Segunda línea del saludo.

Plantilla de código *E6_lee2.php*

```
<?php
// Definir variable cadena con nombre y ruta del archivo
// Usar variable para guardar tamaño del archivo. Hay que usar filesize
// Intentar abrir fichero. Usar operador @ para evitar que se vea error
// Comprobar si se ha podido abrir archivo. A través del descriptor de archivo
// En caso de que el fichero no se haya podido abrir porque no
// exista, mostrar mensaje y salir del programa con exit
// En caso de que el fichero exista, leer todo su contenido con fread
// Comprobar si el fichero tenía datos o estaba vacío.
// Mostrar mensaje en cada caso
// Cerrar archivo
?>
```

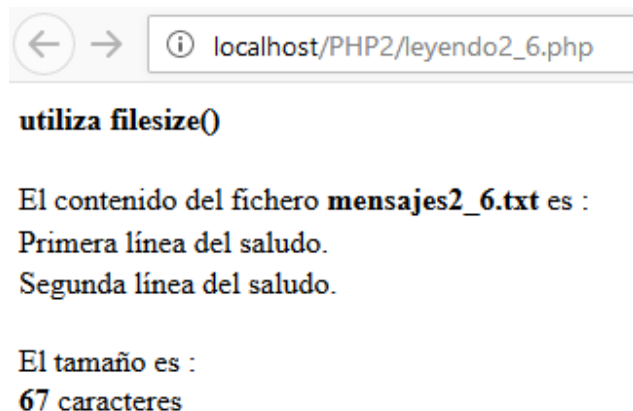
3. Crear archivo *E6_lee3.php*. Se encargará de leer el contenido del archivo *E6_mensajes3.txt* que se ha creado con el bloque de notas

Contenido de *E6_mensajes3.txt*:

**Primera línea del saludo.
Segunda línea del saludo.
**

El programa *E6_lee3.php* es una modificación del ejemplo anterior, pero ahora añade código necesario para que, después de mostrar el contenido del fichero, muestro también el tamaño de este.

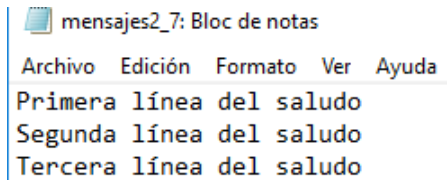
Utilizar la función *filesize ()*.



Plantilla de código:

```
// Definir variable cadena con nombre y ruta del archivo
// Controlar si el fichero existe con file_exists
// En caso de que el fichero exista, mostrar su contenido con readfile
// No hace falta cerrar archivo
```

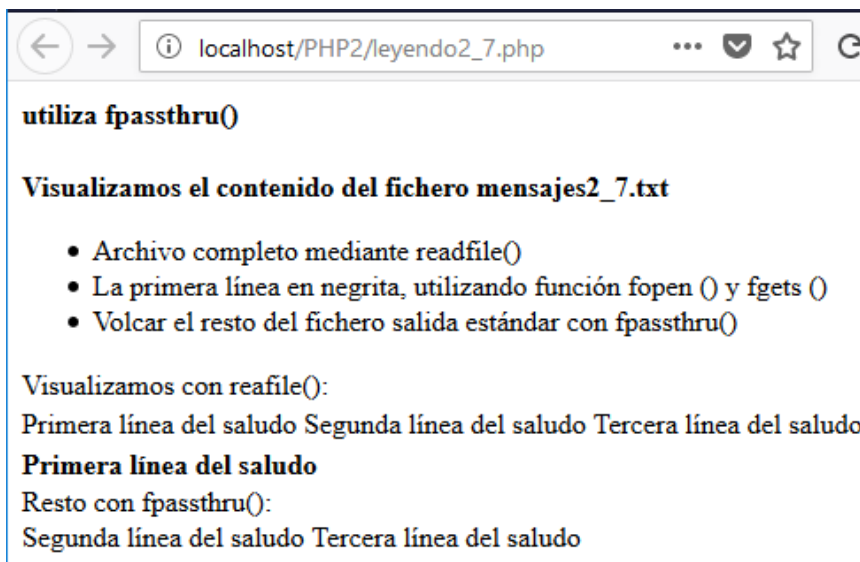
4. Creamos archivo de texto llamado *E6_mensajes4.txt* con el bloc de notas con este formato:



```
mensajes2_7: Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
Primera línea del saludo
Segunda línea del saludo
Tercera línea del saludo
```

El programa *E6_lee4.php* lee el fichero *E6_mensajes4.txt* y muestra su contenido de la siguiente forma:

- Inicialmente se muestra el contenido completo del fichero mediante *readfile()*.
- Después se lee la primera línea en negrita, utilizando para lo cual la función *fopen()* y *fgets()*.
- Volcar el resto del fichero en la salida estándar mediante la función *fpassthru()* .



← → ⓘ localhost/PHP2/leyendo2_7.php ... 🔖 ↻

utiliza fpassthru()

Visualizamos el contenido del fichero mensajes2_7.txt

- Archivo completo mediante *readfile()*
- La primera línea en negrita, utilizando función *fopen ()* y *fgets ()*
- Volcar el resto del fichero salida estándar con *fpassthru()*

Visualizamos con *readfile()*:

Primera línea del saludo Segunda línea del saludo Tercera línea del saludo

Primera línea del saludo

Resto con *fpassthru()*:

Segunda línea del saludo Tercera línea del saludo

Código plantilla:

```
// Mostrar mensajes de presentación
// Definir variable cadena con nombre y ruta del archivo
// Hacer lectura inicial con readfile
// Intentar abrir fichero.
// Comprobar si se ha podido abrir archivo. A través del descriptor de archivo
// Leer primera línea de archivo con fgets
// Leer resto de archivo con passthru
// Se cierra archivo
```

Recuperando peticiones desde archivo

Lectura con readfile

5. Crear el archivo *E6_lee5.php*.

Imaginemos que se tienen guardadas las peticiones de compra en un fichero llamado *E6_mensajes5.txt* con un formato plano.


Estructura del archivo:

FechaCompra =====	Usuario =====	Dirección =====	Frenos =====	Aceite =====	Ruedas =====
----------------------	------------------	--------------------	-----------------	-----------------	-----------------

Nota: Si se realiza la lectura de su contenido mediante *readfile()*, se mostrará sin respetar los cambios de línea finales.

Caso 1: El archivo origen no tiene caracteres `
`:

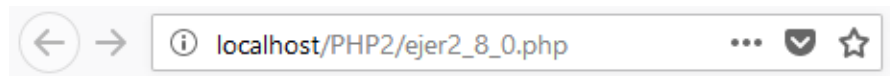
Contenido de este archivo (*E6_mensajes5.txt*):

 ejer2_8_0: Bloc de notas

Archivo	Edición	Formato	Ver	Ayuda
26/01/2018	pepe	C/Colón,45	1	2 3
27/01/2018	pepe	C/Turía,12	4	5 6

Crear archivo *E6_lee5.php* que se encargue de leer el contenido del archivo de datos usando `readfile`

Al ejecutar, se mostraría así:



Lectura con `readfile`

Las peticiones contenidas *ejer2_8_0.txt* hasta la fecha 28/1/2018 son:

26/01/2018 pepe C/Colón,45 1 2 3 27/01/2018 pepe C/Turia,12 4 5 6


Nota:

Si queremos que `readfile` detecte los cambios de línea, hay que introducirlos como código HTML dentro del fichero, es decir, caracteres `
`.

Ver ejemplo de ejecución con *E6_mensajes5_br.txt*:

Caso 2: El archivo origen tiene caracteres `
`:

Contenido de este archivo (*E6_mensajes5_br.txt*):

 ejer2_8_con_br: Bloc de notas

Archivo	Edición	Formato	Ver	Ayuda	
26/01/2018	pepe	C/Colón,45	1	2	3
27/01/2018	pepe	C/Turia,12	4	5	6

Modifica archivo *E6_mensajes5.txt* para que se encargue de leer el contenido del archivo de datos *E6_mensajes5.txt_con_br.txt* usando `readfile`

Al ejecutar, se mostraría así:

Lectura con `readfile`

Las peticiones contenidas *ejer2_8_con_br.txt* hasta la fecha 14/1/2020 son:

26/01/2018 pepe C/Colón,45 1 2 3

27/01/2018 pepe C/Turia,12 4 5 6

Lectura línea a línea con `fgets()`

6. Crear archivo `E6_lee6.php`:

Se encarga de acceder en modo lectura al contenido del fichero `E6_mensajes6.txt`:

Se pretende:


- Utilizar la función `fgets()` para acceder a cada línea del archivo
- Queremos mostrar cada línea del archivo en una línea diferente al visualizar por pantalla
- Usa bucle para detectar final de archivo y dejar de leerlo
- Se muestra mensaje con la fecha de hoy. Usar función `getdate()`.

<https://www.php.net/manual/es/function.getdate.php>

`E6_mensajes6.txt`

Será quedan almacenadas las compras.

Formato

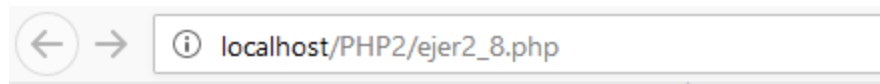
 ejer2_8: Bloc de notas

Archivo	Edición	Formato	Ver	Ayuda		
26/01/2018		pepe	C/Colón,45	1	2	3
27/01/2018		pepe	C/Turía,12	4	5	6

Nota:

Vemos como `gets()` detecta el carácter de final de línea (<return>) y por eso se escribe en líneas diferentes, al contrario que ocurría con `readfile()`, que no los detectaba y lo escribía a continuación.

Salida:



Lee Archivo con fgets

Contenido del fichero **ejer2_8.txt** hasta fecha de hoy (29/1/2018):

26/01/2018 pepe C/Colón,45 1 2 3

27/01/2018 pepe C/Turia,12 4 5 6

Lectura carácter a carácter con `fgetc()`

7. Crear archivo `E6_lee7.php`

Se encargará de realizar la lectura del contenido de un archivo carácter a carácter. El nombre del archivo a leer es `E6_mensajes7.txt`:

Contenido del archivo `E6_mensajes7.txt`:



Utiliza `fgetc()`

Contenido del fichero `ejer2_9.txt` hasta fecha 28/1/2018:

26/01/2018 Pepe C/Colón,45 1 2 3


27/01/2018 Juan C/Turia,12 4 5 6

Condiciones:

- Usar función `fgetc()`
- El archivo de origen tendrá caracteres de cambio de línea ("`\n`"),
- El archivo de origen `E6_mensajes7.txt` tendrá caracteres tabulador ("`\t`") para separar las informaciones de las columnas y el carácter de cambio de línea ("`\n`") para indicar salto de línea.
- Usará bucle para recorrer contenido del archivo
- Comprobará si el carácter leído es el de newline ("`\n`") y lo sustituirá en por un salto de línea en HTML (`
`).

- Cargará los caracteres de cada línea del archivo en una cadena y le añadirá lo
 al final porque al visualizar por pantalla, el navegador lo muestro en líneas separadas

Estructura del archivo de datos::

 ejer2_9: Bloc de notas

Archivo	Edición	Formato	Ver	Ayuda		
26/01/2018		Pepe	C/Colón,45	1	2	3
27/01/2018		Juan	C/Turia,12	4	5	6
28/01/2018		Luís	C/Chile,13	7	8	9

Ejecución:



Utiliza fgetc()

Contenido del fichero ejer2_9.txt hasta fecha 28/1/2018:

26/01/2018 Pepe C/Colón,45 1 2 3

27/01/2018 Juan C/Turia,12 4 5 6

28/01/2018 Luís C/Chile,13 7 8 9

Código plantilla:

```
// Definir variable cadena con nombre y ruta del archivo
// Intentar abrir fichero en modo lectura
// Comprobar si se ha podido abrir archivo. A través del descriptor de archivo
// En caso de que el fichero se haya podido abrir ==> abandonar
// En caso contrario se inicia bucle de lectura por carácter
// Casos diferentes, según sea carácter normal o final línea
// Se cierra archivo
```