

# Programación III

## PHP

### Clase 3

Octavio Villegas

# Temas a Tratar

- Manipulación de archivos
- Envío de datos

# Temas a Tratar

- Manipulación de archivos
  - Generalidades
  - Abrir archivos
  - Cerrar archivos
  - Leer archivos
  - Escribir archivos
  - Copiar archivos
  - Borrar archivos
- Envío de datos

# E/S con Archivos

- El manejo de archivos (de texto o binarios) es una parte importante de una aplicación web.
- PHP nos provee de una extensa gama de funciones de acceso a archivos.
- En esta clase veremos las funciones básicas:
  - fopen (abrir)
  - fclose (cerrar)
  - fread/fgets (leer)
  - fwrite/fputs (escribir)
  - copy (copia)
  - unlink (eliminar)

# Temas a Tratar

- Manipulación de archivos
  - Generalidades
  - Abrir archivos
  - Cerrar archivos
  - Leer archivos
  - Escribir archivos
  - Copiar archivos
  - Borrar archivos
- Envío de datos

# fopen() (1/2)

- Nos permite abrir un archivo ya sea de manera local o externa (http:// o ftp://).
- El primer parámetro de **fopen** contiene el nombre del archivo a ser abierto, y el segundo especifica el modo en que el archivo será abierto.
- El valor de retorno de **fopen** es un entero. Nos servirá para referenciar al archivo abierto.

```
<?php
    int fopen (archivo, modo) ;
?>
```

# fopen() (2/2)

Modo	Descripción
r	Abre un archivo para sólo lectura. El cursor comienza al principio del archivo.
w	Abre un archivo para sólo escritura. Si no existe, crea uno nuevo. Si existe, borra el contenido.
a	Abre un archivo para sólo escritura. Si no existe, crea uno nuevo. Si existe, mantiene el contenido. El cursor comienza en el final del archivo.
x	Crea un nuevo archivo para sólo lectura. Retorna FALSE y un error si el archivo existe.
r+	Abre un archivo para lectura/escritura. Ídem r.
w+	Abre un archivo para lectura/escritura. Ídem w.
a+	Abre un archivo para lectura/escritura. Ídem a.
x+	Crea un archivo para lectura/escritura. Ídem x.

# Temas a Tratar

- Manipulación de archivos
  - Generalidades
  - Abrir archivos
  - Cerrar archivos
  - Leer archivos
  - Escribir archivos
  - Copiar archivos
  - Borrar archivos
- Envío de datos



# fclose()

- Nos permite cerrar un archivo abierto.
- **fclose** requiere el indicador del archivo a ser cerrado (la variable que referencia al archivo).
- Retorna TRUE si tuvo éxito, FALSE caso contrario.

```
<?php
    $ar = fopen(archivo, modo);
    //MAS CODIGO
    fclose($ar);
?>
```

# Temas a Tratar

- Manipulación de archivos
  - Generalidades
  - Abrir archivos
  - Cerrar archivos
  - Leer archivos
  - Escribir archivos
  - Copiar archivos
  - Borrar archivos
- Envío de datos

# fread()

- Nos permite leer de un archivo abierto.
- El primer parámetro de ***fread*** contiene el indicador del archivo a ser leído, y el segundo especifica el número máximo de bytes que serán leídos.
- Retorna un string que representa al contenido del archivo leído.

```
<?php
    echo fread(indicador_archivo,
filesize(archivo));
    //Lee el archivo completo
?>
```

# fgets()

- Nos permite leer una línea de un archivo abierto.
- Requiere como parámetro el indicador del archivo a ser leído, y retorna un string que representa la línea que fue leída.
- Después de cada llamada a ***fgets***, el cursor se mueve a la siguiente línea.

```
<?php
    string fgets(indicador_archivo);
?>
```

# feof() (End Of File)

- Retorna un booleano que indica si se ha llegado al fin del archivo.
- Requiere cómo parámetro el indicador.

```
<?php
    $ar = fopen(archivo, modo);
    //Lee línea a línea hasta EOF
    while(!feof($ar))
    {
        echo fgets($ar), "<br/>";
    }
    fclose($ar);
?>
```

# Temas a Tratar

- Manipulación de archivos
  - Generalidades
  - Abrir archivos
  - Cerrar archivos
  - Leer archivos
  - Escribir archivos
  - Copiar archivos
  - Borrar archivos
- Envío de datos

# fwrite() - fputs()

- Nos permite escribir en un archivo abierto.
- La función parará cuando llegue al fin del archivo o cuando alcance la longitud especificada.
- El primer parámetro contiene el archivo a ser leído, y el segundo es el string a ser escrito. El tercer parámetro es opcional e indica la cantidad de bytes a ser escritos.
- Retorna la cantidad de bytes que se escribieron o FALSE si hubo error.

```
<?php
    fwrite(indicador_archivo, texto [,longitud]);
    fputs(indicador_archivo, texto [,longitud]);
?>
```

# Temas a Tratar

- Manipulación de archivos
  - Generalidades
  - Abrir archivos
  - Cerrar archivos
  - Leer archivos
  - Escribir archivos
  - Copiar archivos
  - Borrar archivos
- Envío de datos



# copy()

- Permite copiar un archivo.
- Los parámetros que recibe son los nombres de los archivos. El primero es el archivo origen, luego el destino de la copia.
- Retorna TRUE en caso de éxito o FALSE si hubo algún error.

```
<?php  
    copy(archivo_origen, archivo_destino);  
?>
```

# Temas a Tratar

- Manipulación de archivos
  - Generalidades
  - Abrir archivos
  - Cerrar archivos
  - Leer archivos
  - Escribir archivos
  - Copiar archivos
  - **Borrar archivos**
- Envío de datos

# unlink()

- Permite eliminar un archivo.
- Recibe el nombre del archivo a ser borrado como primer parámetro.
- Retorna TRUE en caso de éxito o FALSE si hubo algún error.

```
<?php  
    unlink(archivo) ;  
?>
```

# Temas a Tratar

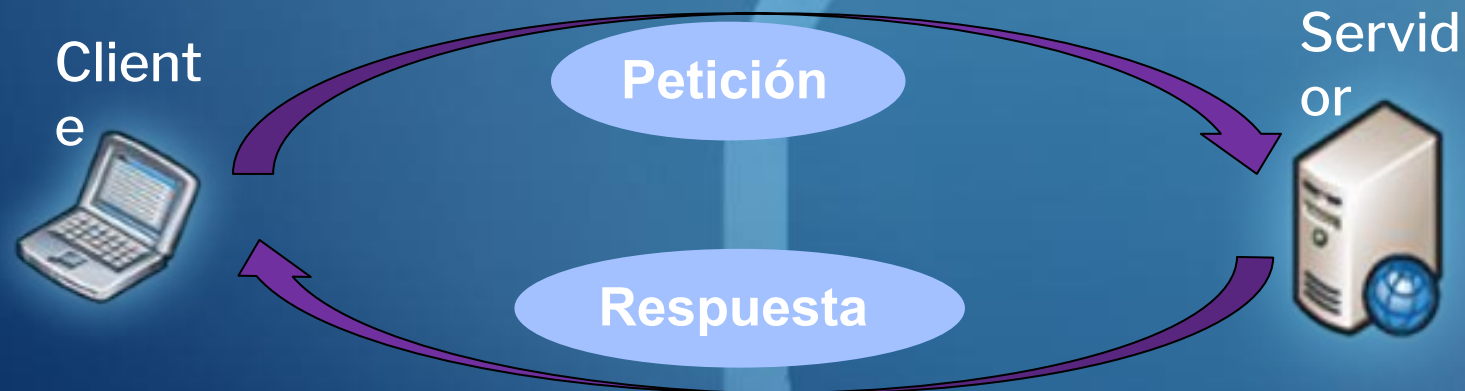
- Manipulación de archivos
- Envío de datos

# Temas a Tratar

- Manipulación de archivos
- Envío de datos
  - HTTP
  - GET
  - POST

# HTTP (Hypertext Transfer Protocol)

- HTTP está diseñado para permitir comunicaciones entre clientes y servidores.
- HTTP funciona como un protocolo de pedido-respuesta entre cliente y servidor.
- Un navegador web puede ser el cliente y una aplicación sobre un computador que aloja un sitio web puede ser el servidor.



# Temas a Tratar

- Manipulación de archivos
- Envío de datos
  - HTTP
  - GET
  - POST

# Método GET

- El par de nombres/valores es enviado en la dirección URL (texto claro).
- Las peticiones GET se pueden almacenar en caché.
- Permanecen en el historial del navegador.
- Pueden ser marcadas (book marked).
- Nunca debe ser utilizado cuando se trata de datos confidenciales.
- Tiene limitaciones de longitud de datos (longitud máxima de 2048 caracteres en la URL).



# Temas a Tratar

- Manipulación de archivos
- Envío de datos
  - HTTP
  - GET
  - POST

# Método POST

- El par de nombres/valores es enviado en el cuerpo del mensaje HTTP.
- Las peticiones POST no se almacenan en caché.
- No permanecen en el historial del navegador.
- No pueden ser marcadas.
- No poseen restricciones de longitud de datos.

# Manejo de Formularios

- Tanto GET como POST crean un *array asociativo*.
- Dicho array contiene pares de clave-valor, dónde las claves son los nombres (atributo ***name***) de los controles del formulario y los valores son la entrada de datos del usuario.
- PHP utiliza las ***super globales*** \$\_GET, \$\_POST y \$\_REQUEST para recolectar datos provenientes de un Form.
- \$\_GET es un array pasado por GET.
- \$\_POST es un array pasado por POST.
- \$\_REQUEST es un array asociativo que contiene \$\_GET, \$\_POST y \$\_COOKIE.



# Ejercitación