

# PWD-Programación Web Dinámica



## PHP – Seguridad



**Universidad Nacional del Comahue**  
**Facultad de Informática**  
**Departamento de Programación**

# Repaso / Variables predefinidas

PHP ofrece un gran número de variables predefinidas a todos los scripts.

- desde variables externas.
- variables incorporadas de entorno.
- últimos mensajes de error.
- las últimas cabeceras recuperadas.

# Repaso / Superglobales

**Superglobales:** variables que están disponibles en todos los ámbitos. Se implementan como arreglos asociativos.

- **`$GLOBALS`** — Referencia todas las variables disponibles en el ámbito global.
- **`$_SERVER`** — Información del servidor y el entorno de ejecución.
- **`$_GET`** — Variables HTTP GET
- **`$_POST`** — Variables HTTP POST
- **`$_FILES`** — Variables de Carga de Archivos HTTP
- **`$_REQUEST`** — Variables de petición HTTP
- **`$_SESSION`** — Variables de sesión
- **`$_COOKIE`** — Cookies HTTP

# \$GLOBALS

- Una variable tipo array asociativo que contiene referencias a todas las variables que se encuentran definidas actualmente en el ámbito global del script. Los nombres de las variables son las claves de la matriz.
- EjemploPHP001GLOBALS.php



# \$\_SERVER

- **`$_SERVER`** es una matriz que contiene información tal como cabeceras, rutas y ubicaciones de scripts. Las entradas de esta matriz son creadas por el servidor web.
- Algunas de las variables mas usadas:
  - `'SERVER_ADDR'` La dirección IP del servidor bajo la cual está siendo ejecutado el script actual.
  - `'SERVER_NAME'` El nombre del servidor anfitrión bajo el que está siendo ejecutado el script actual. Si el script está corriendo en un host virtual, éste será el valor definido para ese host virtual.
  - `REMOTE_ADDR` La dirección IP desde donde el usuario está observado la página actual.
  - `'REMOTE_HOST'` El nombre Host desde donde el usuario está viendo la página actual. La consulta dns de vuelta está basada en el valor `REMOTE_ADDR` del usuario.
- ¿Para que pueden servir estas variables?

# **\$\_SERVER**

- Log de ips que entraron al sitio.
- Ubicación geográfica de las visitas.
- Seguridad por ip.
- En encuestas no dejar votar dos veces a la misma ip.
- Actualización de variables que referencian al servidor.
- .....

# ***\$\_GET, \$\_POST y \$\_REQUEST***

- **\$\_GET**: Una matriz asociativa de variables pasadas al script actual a través del método HTTP GET.
- **\$\_POST**: Una matriz asociativa de variables pasadas al script actual a través del método HTTP POST.
- **\$\_REQUEST**: Un valor tipo array asociativo que contiene de forma predeterminada los datos de **\$\_GET**, **\$\_POST**, y **\$\_COOKIE**

# **\$\_FILES**

- Una variable tipo array asociativo de elementos cargados al script actual a través del método POST.
- Permite acceder a archivos enviados por POST.



# \$\_COOKIE

- Una *cookie* es un archivo creado por un sitio web que contiene pequeñas cantidades de datos y que se envían entre un emisor y un receptor (servidor – navegador)
- Su propósito principal es identificar al usuario almacenando su historial de actividad en un sitio web específico, de manera que se le pueda ofrecer el contenido más apropiado según sus preferencias.
- PHP soporta transparentemente cookies HTTP.

# **\$\_COOKIE**

- Cualquier cookie enviada al servidor desde el cliente, automáticamente se convertirá en una variable PHP igual que ocurre con los métodos de datos *GET* y *POST*.
- Dependen del cliente. El browser puede no permitir que se guarden cookies
- ¿Que se puede guardar en una Cookie?

# **\$\_COOKIE**

- Se puede guardar:
  - Cualquier valor que es de ingreso rutinario como por ejemplo Nombres de usuario.
  - .....
- Ejemplo de uso de cookies:
  - <http://www.desarrolloweb.com/articulos/estilos-css-distintos-a-pagina-php-y-cookies.html>

# Cookie : Métodos

- bool **setcookie** ( Nombre , Valor)
- Se pueden poner cookies usando la función setcookie().
- Las Cookies son parte de la cabecera HTTP, por tanto la función setcookie() debe ser llamada antes de que se produzca cualquier salida al navegador.



# Cookie : Métodos

- **bool setcookie** ( string \$name [, string \$value [, int \$expire [, string \$path [, string \$domain [, bool \$secure [, bool \$httponly ]]]]] )
- *name*: El Nombre de la cookie.
- *Value*: El valor que es guardado en el cliente.
- *Expire*: El tiempo en que expira la cookie en hora Unix. Ej. *time()+60\*60\*24\*30* expira en 30 días.
- *Path*: La ruta en la cual la cookie va a estar disponible. Si no se pone nada, estará disponible para todos.
- *Domain*: El dominio en el cual la cookie va a estar disponible.
- *EjemploPHP003COOKIE.php*

# \$\_SESSION

- Una matriz asociativa que contiene las variables de sesión disponibles en el script actual.
- Las sesiones soportadas por PHP consisten en una forma de preservar ciertos datos a través de los accesos que realiza un usuario en un sitio.
- A cada visitante que accede al sitio, se le asigna un id único que es llamado ***session\_id***. Este es guardado en una cookie en el cliente o es propagada en la URL y es la clave para acceder a los datos de sesión del usuario.
- La sesión permite registrar un número de variables que se mantendrán mientras el visitante acceda al sitio.
- Cuando el visitante accede al sitio PHP puede crear una sesión automáticamente si `session.auto_start` está seteado en 1 o explícitamente a través de la función ***session\_start()***.

# **\$\_SESSION**

- ¿Que se puede guardar en variables de Session?
- Cualquier valor que se necesita en mas de una pagina del sitio.
- Ej. El Nombre de usuario.

# Session : Métodos

- **session\_start()** crea una sesión (o la continúa basandose en el session id pasado por GET o mediante una cookie).
- Si está seteado el inicio automático de sesiones no hace falta llamar a la función.
- **session\_destroy()**: destruye toda la información asociada con la sesión actual.
- **session\_id()** se usa para obtener o establecer el id de sesión para la sesión actual.



# **`$_SESSION`**

- Escribir variables de sesión:  
`$_SESSION["Nombre"] = Valor;`
- Leer variables de sesión:  
`$_SESSION["Nombre"]`
- EjemploPHP004SESSION.php

# Encriptación

- string **md5** ( string \$cadena [, bool \$modo\_binario ] )
- Calcula y devuelve el hash (resumen) MD5 de *cadena* usando el » Algoritmo de Resumen de Mensajes MD5 de RSA Data Security, Inc. . El hash devuelto consiste en una cadena de 32 números en formato hexadecimal. Si el valor del parámetro *modo\_binario* es **TRUE**, el resumen md5 se devuelto tiene una longitud de 16 y su formato es binario.
- ¿Para que sirven los algoritmos de encriptación?

# Encriptación

- ¿Para que sirven los algoritmos de encriptación?
- ¿Cuál es el peligro de encriptar en el servidor?
- Se puede encriptar directamente en el cliente con js.
- EjemploPHP005MD5.php

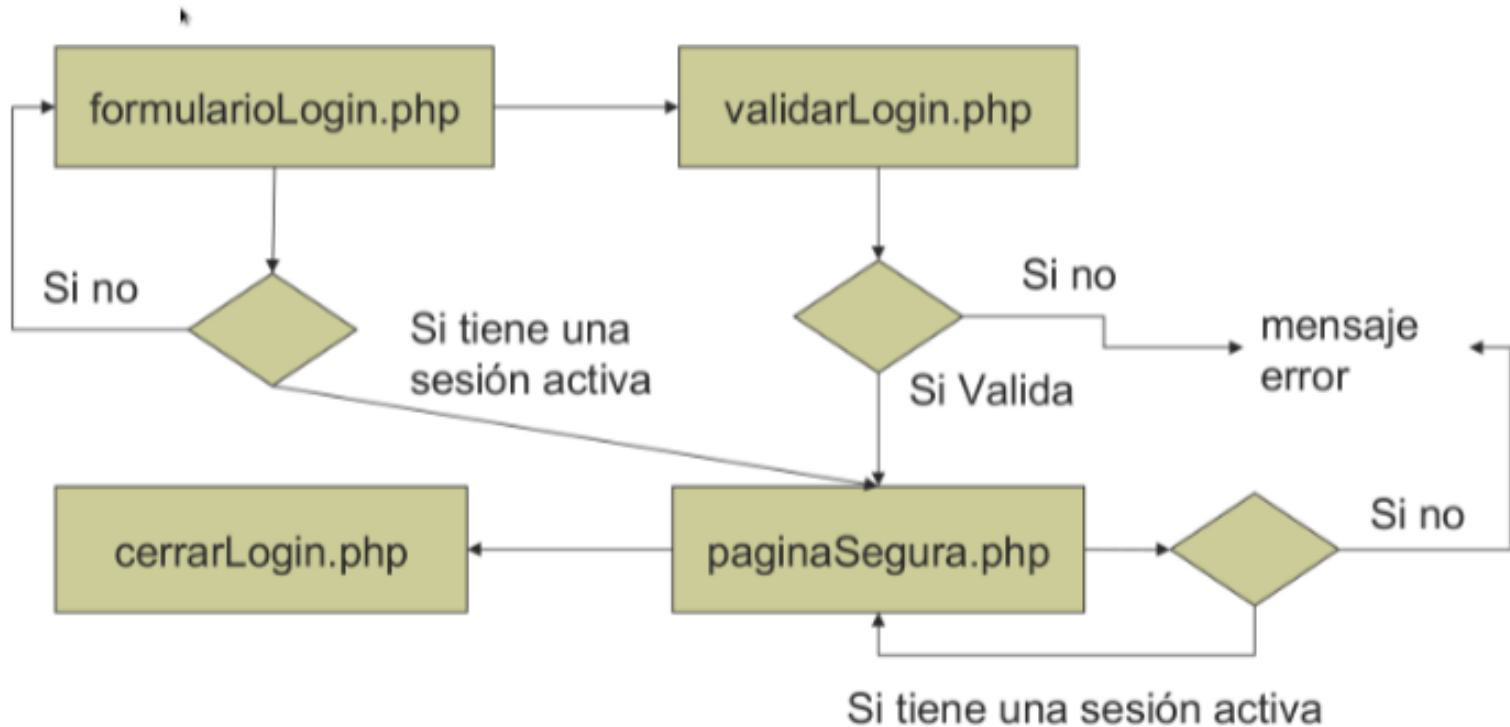
## Uso de variables de sesion Autenticación

La autenticación de un usuario, consiste en verificar su identidad.

En las aplicaciones web comúnmente se utiliza un login con algún dato que identifique a la persona como: nombre de usuario/dni/dirección de mail... y una clave.



# Actividades Login



# Función header

*void header ( string \$string [, bool \$replace [, int \$http\_response\_code ] ] )*

- es usada para modificar el header http.
- debe ser llamada antes de realizar cualquier salida.
- se puede utilizar para redireccionar a otra página.

```
header('Location: http://www.example.com/');
```

# Referencias y Links

- Manual
  - <http://php.net>