

**title:** Clase 2 Proyecto 2015  
**Author:** Einar Lanfranco, Claudia Banchoff  
**description:** Accesibilidad - Procesamiento en el servidor - PHP  
**keywords:** Servidor Web, PHP, Configuración  
**css:** proyecto.css

---

## Proyecto de Software

### Cursada 2016

---

**data-rotate:** 270

### Hoy seguimos con ...



### Primero un repaso

---

### Repaso: Estilos

- Permiten definir los aspectos de visualización

- Ejemplo: [proyecto.css](#)
- 

## Repaso: Estilos

### Importante

- Utilizar los elementos HTML según el objetivo para lo que fueron creados.
  - Usar HTML para la estructura del documento y CSS para la visualización.
  - Usar validadores para revisar estándares.
- 

## Accesibilidad web

---

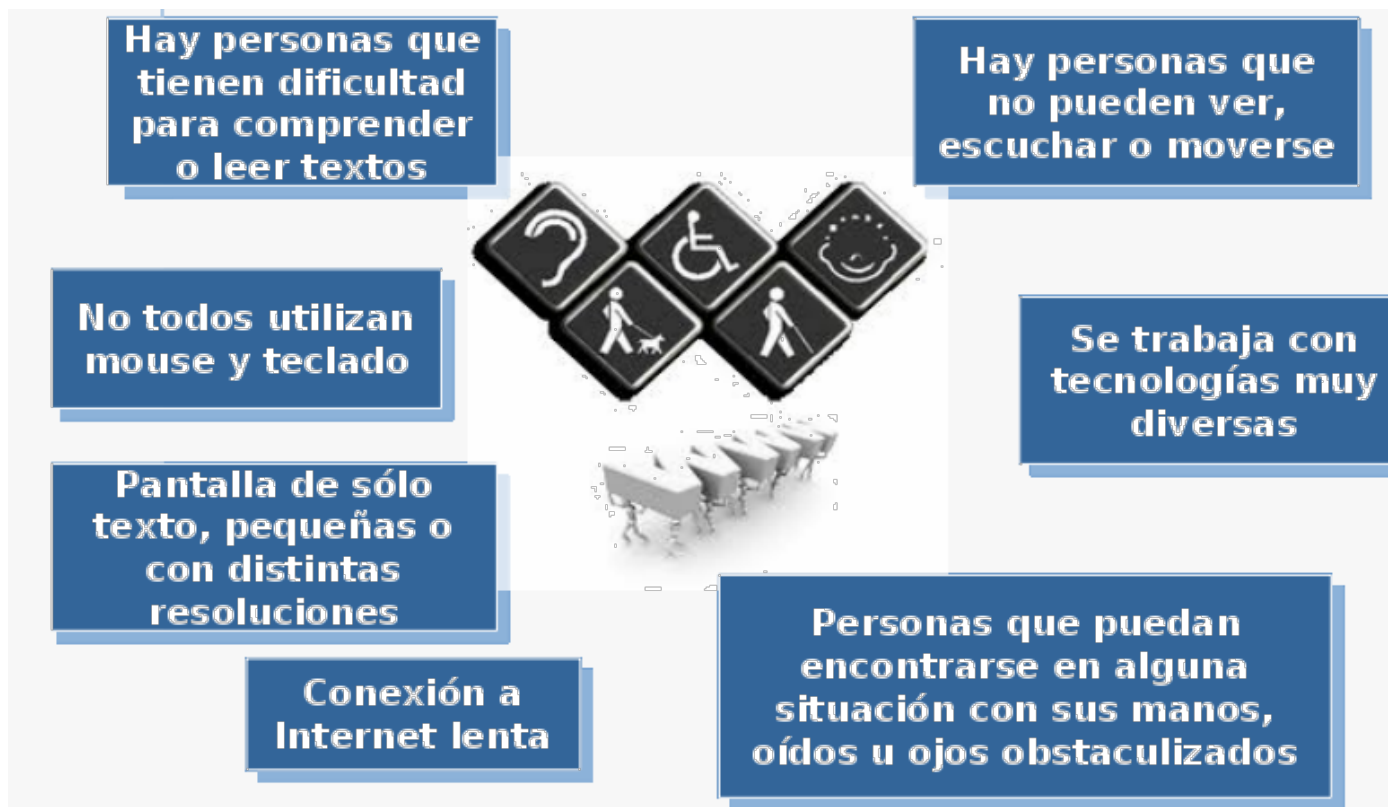
**class:** destacado

## Accesibilidad web

"The power of the Web is in its universality. Access by everyone regardless of disability is an essential aspect".

- Tim Berners-Lee, W3C
-

## Algunas reflexiones



---

## Algunas reflexiones

- Datos [diciembre-2015](#)
  - Según la OMS, alrededor del 15% de la población mundial, o sea más de **MIL millones de personas**, padece alguna forma de discapacidad.
  - Entre 110 millones y 190 millones de adultos tienen dificultades considerables para funcionar.
- Datos [agosto-2014](#)
  - Aproximadamente 285 millones de personas posee discapacidad visual, 39 millones son ciegos y 246 millones presentan baja visión.

---

**class:** destacado

# WAI - Web Accessibility Initiative

## Objetivos:

Desarrollar estrategias, pautas, recursos para hacer la web accesible a **personas con discapacidad**. Pero también será accesible en **otros entornos y aplicaciones**, como navegador de voz, teléfono móvil, PC de automóvil. Y ante limitaciones bajo las que opere, como entornos ruidosos, habitaciones infra o supra iluminadas, entorno de manos libres.

---

## WAI - Web Accessibility Initiative

- Para aplicaciones de usuario: **WAI-User Agent**
  - Para herramientas de autor: **WAI-Autoools**
  - Para contenido web: **WCAG**
    - WCAG 1.0
    - **WCAG 2.0**
- 

## Contenido web accesible: Guías WCAG

- Definen **principios de diseño** web.
  - Cada principio tiene **pautas**.
  - Cada pauta tiene **criterios testeables**.
- 

## Validadores de accesibilidad

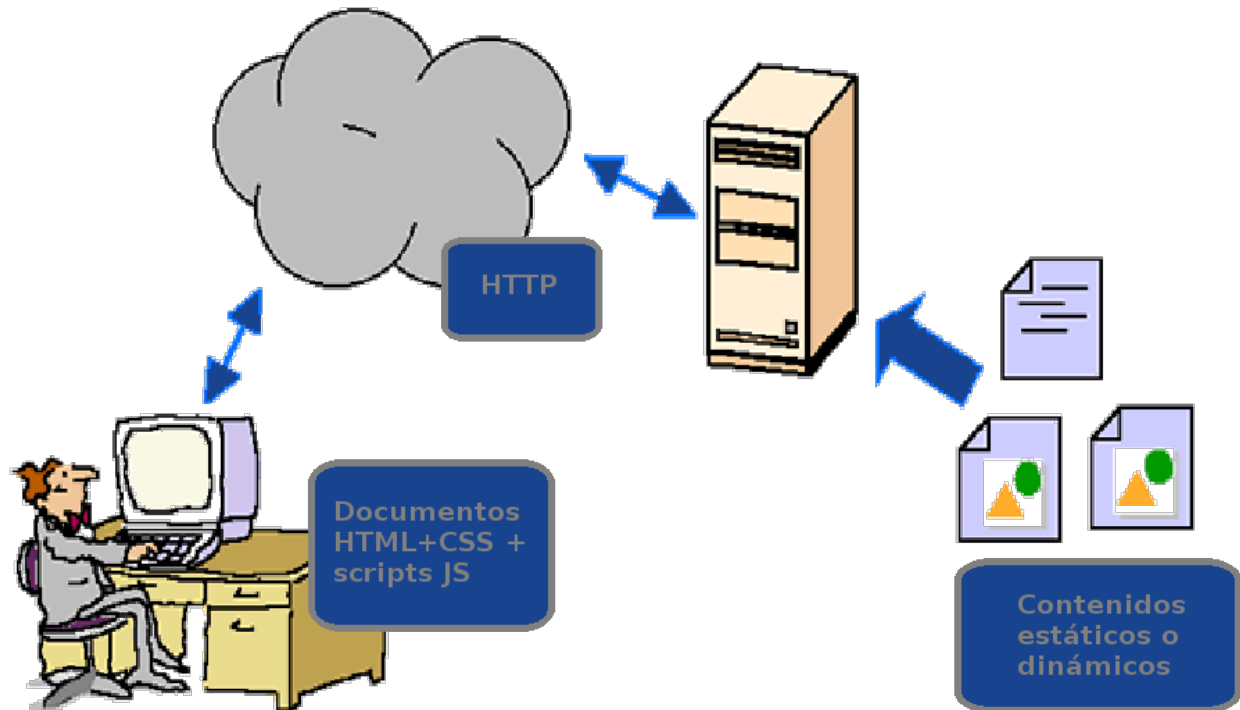
- <http://www.tawdis.net>
  - <http://wave.webaim.org>
  - <http://examinator.ws>
  - **Simor**: tesina de Gaspar Rajoy y Magalí Defalco:
    - <http://simor.linti.unlp.edu.ar/>
- 

## En Argentina

- **Ley Nacional 26.653: Acceso a la Información pública**
- **Sancionada**: noviembre de 2010
- **Reglamentada**: abril de 2013

---

## Server Side

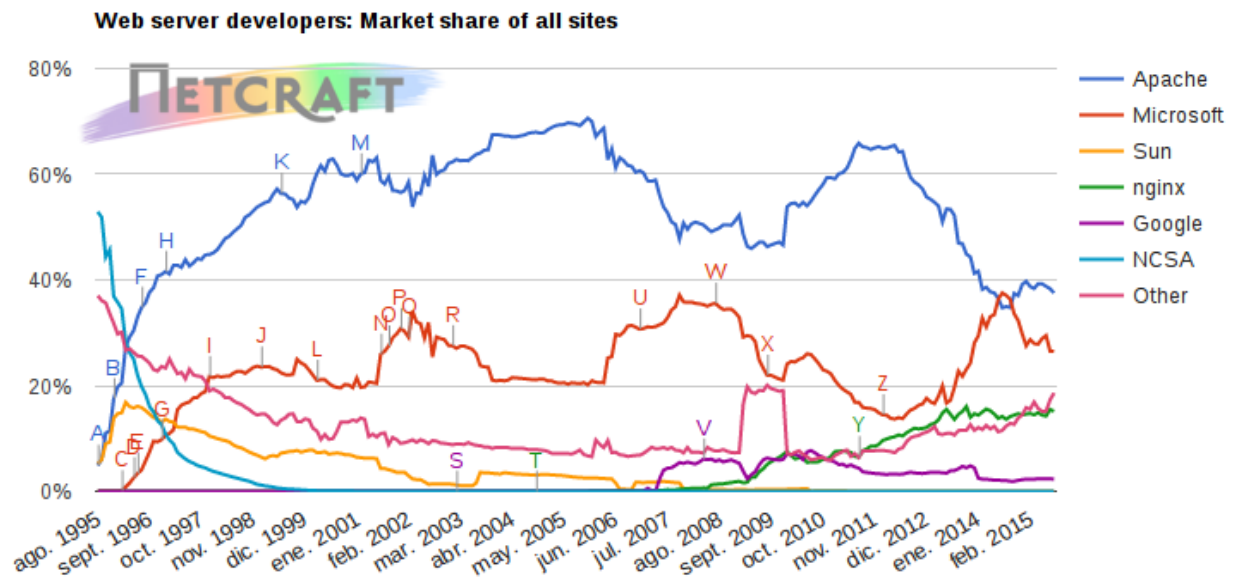


---

## Páginas web

- Estáticas
    - Sólo HTML + CSS
  - Dinámicas
    - Con programación en el cliente
      - Scripts Javascript, etc.
    - Con programación en el servidor
      - Programación CGI/FastCGI
      - Lenguajes de script "server-side": [PHP](#), [Ruby](#), [Perl](#), [Python](#), Java, etc.
-

# Servidores Web



Developer	July 2015	Percent	August 2015	Percent	Change
Apache	325,696,514	38.34%	327,985,968	37.51%	-0.83
Microsoft	225,282,713	26.52%	231,429,146	26.47%	-0.05
nginx	131,460,063	15.47%	132,443,391	15.15%	-0.33
Google	20,255,424	2.38%	19,933,095	2.28%	-0.10

<http://news.netcraft.com/archives/category/web-server-survey/>

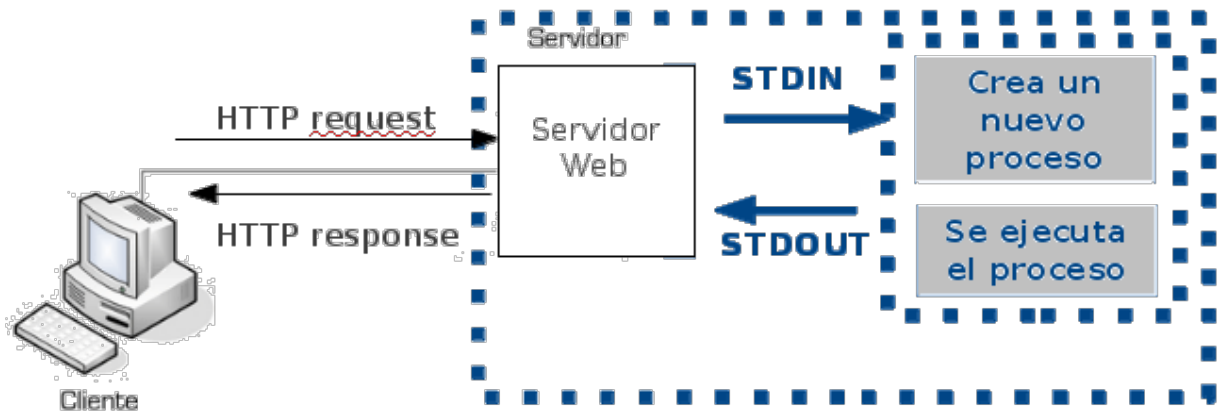
## Procesamiento en el servidor

Existen varias alternativas...

- CGI “Comon Gateway Interface”.
- FastCGI.
- SAPIs (Server Application Programming Interface)
- Lenguajes de Script.

## CGI – RFC 3875

El programa recibe sus datos de la entrada estándar (stdin) a través del CGI



- Ventajas
    - Simple, se puede usar cualquier lenguaje.
  - Desventajas
    - Sobrecarga en la creación de procesos.
    - Problemas de seguridad y portabilidad.
- 

## Fast-CGI

- Plantea una mejora a CGI.
- Nace para mejorar problemas de performance de CGI.

## SAPIs

- Extienden la funcionalidad del servidor.
  - Ejemplo: ISAPI/ Módulos Apache
  - **Ventajas**
    - Mayor performance
  - **Desventajas**
    - Dependientes del servidor
- 

## Herramientas

### ¿Qué necesitamos para arrancar?

- Servidor web: Apache/Nginx
- Intérprete PHP
- La base de datos: MariaDB (- MySQL)

- Algunas soluciones:
    - Soluciones LAMP/WAMP: Linux/Windows - Apache - MariaDB - PHP/Perl/Python....
    - Soluciones LEMP: Nginx x Apache
  - Algunas herramientas:
    - EasyPHP
    - XAMPP
    - etc...
- 

## Nosotros, ¿qué vamos a usar?

- Lenguaje: PHP 5.6.24
  - Servidor Web: Apache 2.4.10
  - Servidor de Base de Datos: MariaDB 10.0.26
- 

## Algunas configuraciones en el servidor web

- **ServerRoot:** Indica dónde se encuentra instalado servidor web y a partir de ahí los archivos de configuración, de logs, etc.
  - **Listen:** Puerto donde se atenderán las peticiones. Por defecto se utiliza el puerto 80 de TCP. También permite especificar qué direcciones IP se utilizarán (por si el servidor tuviese más de una). Ejemplo: Listen 12.12.22.12:80
- 

## Algunas configuraciones en el servidor web

- **DocumentRoot** o **root:** Directorio raíz de nuestro sitio. Generalmente se encuentra en archivo default u otro nombre en el directorio "sites-available".
  - **DirectoryIndex** o **index:** Permite indicar qué página se cargará por defecto en el caso de que no se especifique ninguna en la URL.
  - **ErrorDocument:** establece la configuración del servidor en caso de error.
- 

## PHP

## Hypertext Pre-Processor

---



# Características generales

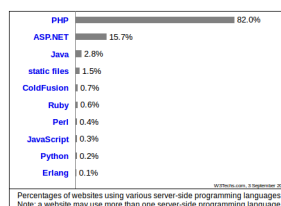
- PHP es un lenguaje de scripting de propósito general que tiene una gran comunidad de usuarios.
- Se utiliza especialmente para aplicaciones web pero puede utilizarse para desarrollar cualquier tipo de aplicación (**ver ejemplosClase2/comocli.php**)
- Es interpretado.
- Es open source distribuido bajo una licencia libre similar a la de BSD, la PHP License v3.01.
- Website: <http://php.net/>
- ¿PHP 5 o PHP 7?

- Ver en <https://inicio.proyecto2016.linti.unlp.edu.ar/>
- 

## PHP en aplicaciones web

- Es **server-side**.
  - Los scripts están embebidos en el código HTML.
  - Permite construir páginas dinámicas según la solicitud del cliente y según la información disponible en el servidor.
  - Se puede correr con la mayoría de los servidores web conocidos (como CGI/FastCGI/[FPM](#)/módulo del servidor).
  - Es independiente de la plataforma donde corre.
  - Tiene un soporte muy amplio para base de datos.
  - Provee soporte para programación orientada a objetos.
- 

## PHP en aplicaciones web

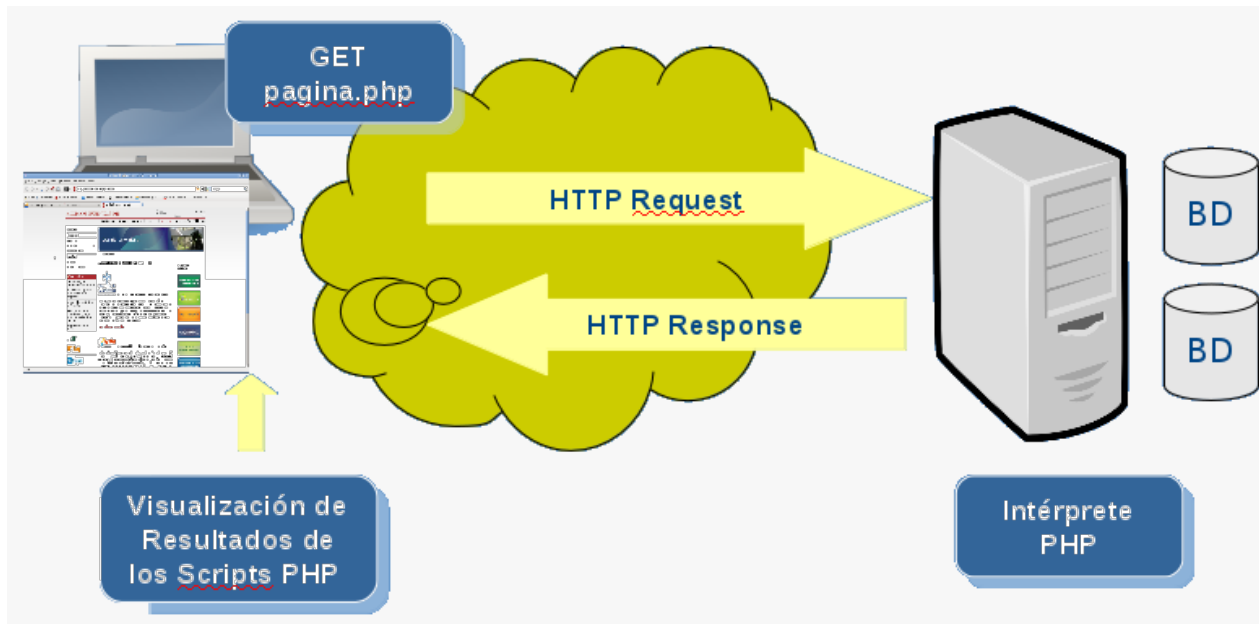


[http://w3techs.com/technologies/overview/programming\\_language/all](http://w3techs.com/technologies/overview/programming_language/all)

---

# PHP en aplicaciones web

## Modelo de ejecución



---

## Configuración: php.ini

- Se lee cuando arranca PHP.
- Algunas opciones se pueden modificar usando la **función ini\_set**.
- La localización por defecto de php.ini es definida en tiempo de compilación.
- Algunas consideraciones:
  - Las líneas que comienzan con ; se ignoran.
  - [php] delimitan una sección – también ignorado.
  - Los valores lógicos son: true, on, yes, false, off, no, none.
- Ejemplo:
  - register\_globals = off
  - track\_errors = yes

---

## php.ini

- Mostrar los errores al cliente:
  - display\_errors = On
- ¿Qué errores mostrar?

• error\_reporting = E\_ALL | E\_STRICT  
**Nota:** E\_STRICT Antes de PHP 5.4 no estaba en E\_ALL

---

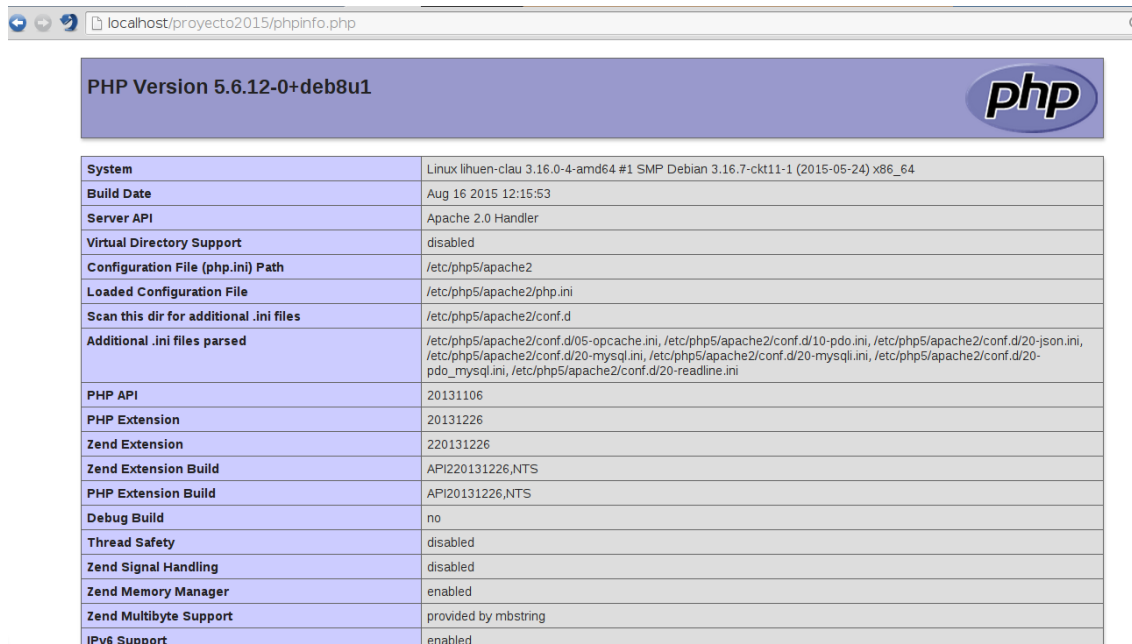
## php.ini

- Mostrar los errores al arranque:
  - display\_startup\_errors = On
- Poner los errores en un archivo:
  - log\_errors = On
  - error\_log = php\_errors.log

Más info: <http://www.php.net/manual/en/errorfunc.configuration.php>

---

## Mi primer script PHP



PHP Version 5.6.12-0+deb8u1	
System	Linux lihueu-clau 3.16.0-4-amd64 #1 SMP Debian 3.16.7-ckt11-1 (2015-05-24) x86_64
Build Date	Aug 16 2015 12:15:53
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/05-opcache.ini, /etc/php5/apache2/conf.d/10-pdo.ini, /etc/php5/apache2/conf.d/20-json.ini, /etc/php5/apache2/conf.d/20-mysql.ini, /etc/php5/apache2/conf.d/20-mysqli.ini, /etc/php5/apache2/conf.d/20-pdo-mysql.ini, /etc/php5/apache2/conf.d/20-readline.ini
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226
Zend Extension Build	API220131226,NTS
PHP Extension Build	API20131226,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled

## PHP: Lo básico

- Existen constantes predefinidas:
  - PHP\_VERSION: la versión de PHP utilizada.
  - PHP\_OS: el sistema operativo sobre el cual se está ejecutando PHP
  - etc.

- Variables predefinidas (superglobals)

- \$GLOBALS, \$\_SERVER, \$\_GET, \$\_POST, \$\_COOKIE, \$\_REQUEST, \$\_SESSION, etc.
  - Ejemplo: Para obtener DOCUMENT\_ROOT se usará **\$\_SERVER['DOCUMENT\_ROOT']**
- 

## Variables externas

Ejemplo: Acceso a los campos de un formulario.

```
<form method="get" action="proceso_formu.php">
  <label for="nombre">Nombre</label>
    <input type="text" name="nombre">
  <label for="apellido">Apellido</label>
    <input type="text" name="apellido">
  <label for="turno">Turno de teoría:</label>
    <select name="turno">
      <option>Martes</option>
      <option>Viernes</option>
    </select>
  <input type="submit" value="Enviar">
</form>
```

---

## \$\_REQUEST, \$\_GET, \$POST

Veamos el [ejemplo](#):

**class:** destacado

---

## \$\_REQUEST, \$\_GET, \$POST

Usemos las variables externas adecuadas según los requerimientos de la aplicación.

---

## Usando archivos

- Existen varias funciones disponibles para manejar archivos en PHP.
- Algunas funciones útiles:
  - fopen(): Abre un archivo o URL
  - filesize(): Obtiene el tamaño del archivo (en bytes)
  - fileperms(): Obtiene los permisos del archivo
  - file\_exists(): Verifica si un archivo existe

- `fgets()`, `fputs()`: Lee/Escribe una línea a un archivo.
  - `fread()`, `fwrite()`....
  - etc.
- 

## Usando archivos

Un [contador](#) de visitas sencillo

---

## Include y require

- **`include()`** y **`require()`**: Permiten incluir y evaluar un archivo dentro de un script.
- Si la inclusión ocurre dentro de una función, todo el código contenido en el archivo incluido se comportará como si hubiese sido definido dentro de esta función.
- Cuando se incluye un archivo, el intérprete sale del modo PHP y entra en modo HTML al principio del archivo referenciado, y vuelve de nuevo al modo PHP al final.
- La diferencia entre estas dos funciones es que **`include()`** produce un warning mientras que **`require()`** produce un error fatal.
- `require_once()`, `include_once()`

Veamos un ejemplos de [include](#) y de include once

---

## Cookies

- Básicamente, son “tokens” en el requerimiento HTTP que permite identificar de alguna manera al cliente en el servidor.
  - Se almacenan en el cliente.
  - Muy usado por ser HTTP un protocolo sin estado.
  - Formato: **`nombreCookie=valor;expires=fecha;`**
  - PHP las considera variables externas: Usa **`$_COOKIE`** (un arreglo con las cookies generadas).
  - Mediante la función `setcookie()` es posible grabar cookies en el cliente.
  - Veamos un ejemplo de uso de [cookies](#)
- 

**class:** destacado

## Sesiones

Es un mecanismo para conservar ciertos datos a lo largo de varios accesos.

- Permite registrar un número arbitrario de variables que se conservarán en las siguientes peticiones.

- Identificador: A cada visitante se le asigna un identificador único, llamado **session id** (identificador de sesión).
  - Hay dos formas de propagar un identificador de sesión:
    - Mediante cookies
    - A través de la URL.
- 

## Sesiones

- Algunas funciones:
    - **session\_start()**: Inicializa los datos de una sesión.
    - **session\_destroy()**: Destruye los datos guardados en una sesión.
    - **session\_id()**: recupera o modifica el valor de la sesión.
  - **Debe ubicarse antes que cualquier salida al browser.**
  - Si no existe un **session ID**, session\_start() lo crea e inicializa un arreglo vacío denominado \$\_SESSION.
- 

## Sesiones

- Los archivos con los datos de la sesión se generan en el servidor.
- El cliente sólo guarda el session Id

Veamos el ejemplo de [sesiones](#)

---

## Referencias

- Curso de accesibilidad de la Facultad: Pedir usuario y clave
- PHP: <http://php.net/docs.php>