

CLASE 09

CONCEPTOS INICIALES

- Sesiones de usuario
- Identificando scripts para procesos
- Variable global `$_SESSION`
- Abriendo la *session*
- Destruyendo la *session*
- `$_SESSION` con funciones de apertura y cierre de sesión

OBJETIVOS

- Usar las variables súper globales aprendidas.
- Comprender el tratamiento de sesiones de usuario.
- Aprender a validar y restringir datos según lo ingresado por el navegante.
- Usar los scripts de ejemplo y ejercicios ubicados dentro del archivo Clase_09_Sesiones.zip
- Aplicar conocimientos de clases anteriores para integrar con estas estructuras.
- Desarrollar código para ser visualizado en el navegador.

DESARROLLO

Introducción

En esta clase seguimos trabajando con **formularios**, con el manejo de las **variables** y sus **valores** que viajan; en esta oportunidad, vamos a **limitar** la visualización de cierta **información**. Esto se realizará mediante el uso de **Sesiones de Usuario**.



`$_SESSION`

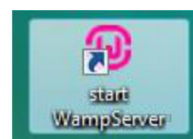
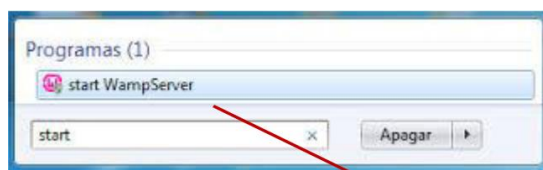
Como todas las clases, ya sabemos cómo empezar a trabajar:



Recordemos los pasos previos para todas nuestras prácticas:

a) Debemos tener activo el WampServer:

Para comenzar a probar recuerden que el servidor local debe estar activado [en verde], tras doble click en el icono del Wamp del escritorio, o tras buscarlo en los Programas instalados:



- b) Abriremos nuestro editor de texto, para trabajar con el código de los ejemplos que iremos viendo a continuación.



- c) Tendremos que descargar el archivo **Clase_09_Panel_Session.zip** y ubicar su contenido en el directorio de alojamiento, creando las carpetas correspondientes para que quede de esta manera.
- d) Abriremos el navegador [Chrome recomendado] y nos ubicaremos en nuestro server y la carpeta recién creada. De allí podremos ir revisando los ejemplos.

http://localhost/Clase_09/Panel_Session

Recordemos nuevamente:



La **ruta física a los archivos** o scripts de nuestra **carpeta de alojamiento de nuestro servidor** (local), será **C:\wamp\www**

La **ruta para ejecutar** esos archivos o scripts, alojadas **en nuestro servidor** (local), será **desde el navegador**, indicando primero el **servidor a donde apuntamos**, y luego las carpetas hasta llegar al **script**: http://localhost/Clase_xx/carpeta....

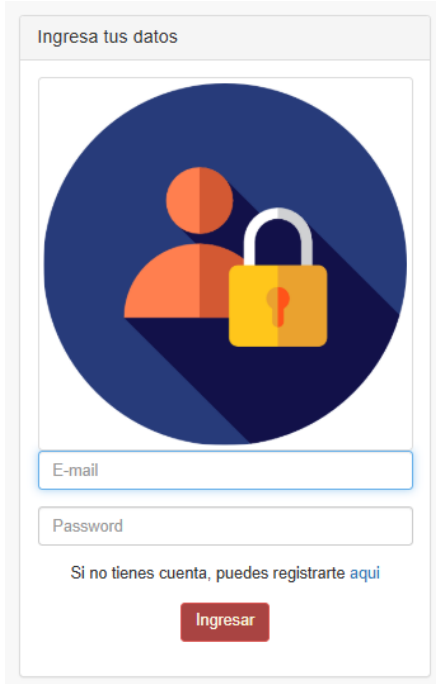
- Noten que envolvemos nuestro código con las etiquetas **<?php** y **?>**
- La sentencia **echo** es la que muestra en pantalla el mensaje que se escribe entre comillas [dobles o simples].
- Recordar siempre terminar las sentencias con punto y coma ;
- Cada **cambio** que se genere en el **script** desde el editor de texto, **guardarlo**. Para ver el cambio en el **navegador**, **refrescar la página** con F5.¹
- Cada script se **guarda con la extensión .php** para que pueda ser interpretado correctamente.

Ahora si, seguimos adelante!

¹ Deshabiliten el caché para que el navegador fuerce a traer los cambios recientes en cada cambio que hagan:
<https://amerikanistik.org/es/software/930-how-to-completely-disable-cache-in-google-chrome.html>

Sesiones de Usuario

Es muy común que trabajando con aplicaciones y sistemas web, exista información que pueda ser expuesta a un **usuario mediante una clave** de acceso, o que para **habilitar cierta sección** se lleve a cabo algún proceso para **validar** los datos del **usuario** que accede, tras lo cual se permite habilitar, o no, ciertas secciones con **información privada**.



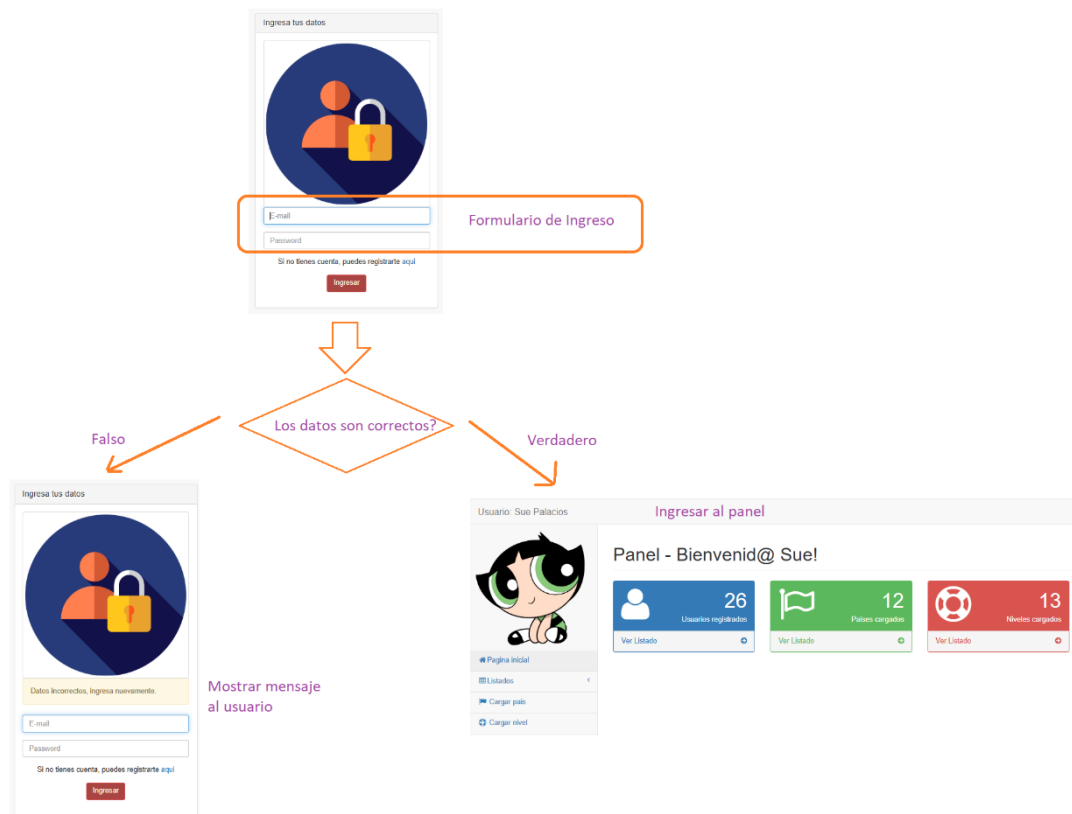
Estos procesos de **login** y **validaciones**, son muy frecuentes que los tenemos presentes siempre cuando queremos ingresar por ejemplo a cualquier sitio donde se deba mostrar información privada. Ejemplos: una casilla de *mail*, el *muro* de facebook, el HomeBanking, es decir toda aplicación donde se haya generado un **usuario con clave** para un **acceso personal**.

Por supuesto que deberemos **ingresar** a estos **paneles** con los datos solicitados de **manera correcta**, de lo contrario veremos el **mensaje de error** respectivo, y hasta en algunos casos puede que se bloquee el acceso si intentamos más de tres veces fallidas, como por ejemplo en un *Home Banking*, como medida de

mayor seguridad.

Una vez que el usuario ha ingresado de **manera correcta**, ya puede visualizar su **panel**, sus **configuraciones** generadas, sus **datos personales**. Con los permisos que cuente podrá ingresar a ciertas secciones, y a otras seguramente estará restringido, a menos que sea un *súper usuario* con acceso global. Todo este panel será navegado por el usuario que accedió con las credenciales correctas, y mientras ese **usuario** se encuentre **dentro del panel**, se considera que está en su **Sesión de Usuario**. Es decir, la sesión se considera como el tiempo que un usuario se relaciona con una aplicación. **La sesión comienza cuando el usuario ingresa, y finaliza cuando cierra su sesión y sale de ella.**

Veamos este procesamiento mediante un diagrama:



Identificando scripts para procesos

Vamos a identificar los diferentes scripts que vamos a ir manejando.

1. El formulario de login inicial: C:\wamp\www\Clase_09\Panel_Session\login.php

- En el navegador http://localhost/Clase_09/Panel_Session/login.php
- El proceso básico para ingresar requerirá los controles básicos: *Cuadro de texto para la cuenta, la clave y botón tipo submit.*
- Una sección con *mensaje si los datos ingresados son incorrectos.*
- Si ingresan datos en los controles de texto y pulsan el botón de **Ingresar**, allí es cuando se debe generar el proceso para el Login.
- Es en este momento, al pulsar el botón, se deben:
 - Validar los datos de usuario y clave. Esto luego será realizado contra la base de datos.
 - Si ambos son correctos, deberá generarse la sesión de usuario, y derivar al usuario al panel.
 - De lo contrario, si los datos fueran incorrectos, deberá quedarse en el login, con el mensaje de error respectivo.

2. La **Bienvenida** al **panel**: C:\wamp\www\Clase_09\Panel_Session\index.php
- En el navegador http://localhost/Clase_09/Panel_Session/index.php
 - Este script contendrá lo que queremos que vea el usuario apenas ingresa a su panel. Noten que hay secciones:
 - Una banda superior, que en la esquina superior izquierda identifica al **usuario activo**, y una imagen.
 - En la esquina superior derecha permite **acciones propias del usuario activo**, como ver Mi Perfil, o Salir.
 - A la izquierda de la sección central existen menús, **acciones permitidas** al usuario logueado.
 - Sección **central** con el **contenido** que deseamos mostrar en esta **bienvenida**. Noten que se identifica por el Nombre **personal** del usuario, es decir contamos con sus **datos** en esta página.

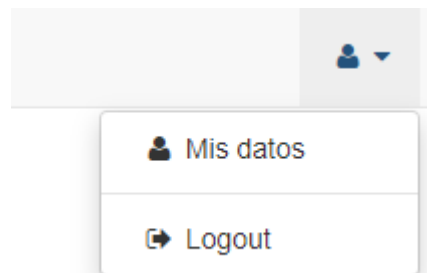


Este **contenido** tal como está, **no debería ser accedido directamente**, es decir, este script al cargarse **deberá validar** si el **usuario** está **logueado** para mostrarse así. De lo contrario, **si** el usuario **no** ha iniciado su **sesión**, deberá **redireccionarlo al script del Login**, para que lo haga.

Pueden ir identificando dentro de cada script las diferentes secciones mencionadas, verán que hay un seccionado usando las funciones `require_once()`.

1. Menú de Opciones del usuario: Esta sección a continuación se encuentra en el [user.inc.php](#).

Contiene las acciones propias del usuario en su sección de la esquina superior derecha: Por ejemplo, ver sus datos personales (veremos el proceso mas adelante), y el “Logout” (Salir).



```
<li class="dropdown">
  <a class="dropdown-toggle" data-toggle="dropdown" href="#">
    <i class="fa fa-user fa-fw"></i> <i class="fa fa-caret-down"></i>
  </a>
  <ul class="dropdown-menu dropdown-user">
    <li><a href="misdatos.php"><i class="fa fa-user fa-fw"></i> Mis datos</a>
    </li>
    <li class="divider"></li>
    <li><a href="cerrarsesion.php"><i class="fa fa-sign-out fa-fw"></i> Logout</a>
    </li>
  </ul>
  <!-- /.dropdown-user -->
</li>
```

3.a) El cierre de sesión al Salir.

Aquí será necesario trabajar el **cierre de la sesión activa**.

Cuando el usuario **pulse en “Salir”**, se deberá apuntar a un script tal que éste cierre la sesión actual, **limpie las variables de sesión** utilizadas [para que otro usuario que ingrese no vea el contenido del último] y **ubique** al navegante en el **login.php**.

Como ven, el link apunta al script “cerrarsesion.php”.

Al llegar allí, se solicita la función para trabajar con las `$_SESSION` (línea 2, `session_start()`). Luego se **limpia el array** de la variable global `$_SESSION` propiamente. Luego la **función para destruir efectivamente** todo rastro de sesión (`session_destroy()`). Una vez todo limpio y sin datos de usuario, redirigimos al login en las últimas dos líneas con **el header(“Location: donde nos dirigimos”)** y fin de proceso actual.

```

1  <?php
2  session_start();
3  $_SESSION=array(); //limpia el array de session
4  session_destroy(); //destruye
5
6  header('Location: login.php');
7  exit;
8
9  ?>

```

2. El resto de scripts que se ven en el menú lateral de la izquierda: serán páginas que deben ser vistas dentro del panel, **no serían públicas**, por lo que en cada una se debe validar que la sesión de usuario esté activa para poder ser visualizada.



Para esto, se deberá colocar en cada página que deba mantenerse sesionada dentro del panel, este bloque de validación:

```

1  <?php
2  session_start();
3
4  //si tengo vacio mi elemento de sesion me tiene q redireccionar al login..
5  //al cerrarsesion para que mate todo de la sesion y el se encarga de ubicar en el login
6  if (empty($_SESSION['Usuario_Nombre']) ) {
7      header('Location: cerrarsesion.php');
8      exit;
9  }
10

```

La condición deberá expresar que al menos contemos con un valor en algún elemento de la \$_SESSION. (línea 6).

Recuerden que esto se seteó tras el login correcto, allí le dimos contenido a esta variable global.

Por lo tanto, si estuviera vacía, quiere decir que no viene de un login válido, por esto hay que limpiar todo rastro y alojar nuevamente en el login, para que se loguee correctamente, y tome las variables de sesión de manera correcta.



Recuerden que podrán encontrar muchas maneras de programar estos procesos, quizás con otros scripts utilizados, programado de otra manera, y de seguro funcionará y podrán concretar los procesos. Aquí se menciona una manera, una bien práctica para seguir, según mi criterio. Ustedes pueden usar la que deseen, siempre y cuando sepan dónde ubicar cada proceso y sepan por qué hacerlo de la forma que eligen.

Habiendo analizado los **procesos** que necesitamos trabajar, y las ubicaciones donde podemos llegar a hacerlo, estamos en condiciones de generar un panel con sesión de usuario tras un acceso correcto.

Variable global \$ SESSION

La variable super global **\$SESSION** es propia de PHP, de tipo **array**, (como las que ya vimos \$_POST y \$_GET), disponible desde cualquier script que sea solicitada, siempre y cuando hayamos abierto una sesión de usuario, y le hayamos cargado valores. Esta apertura debe hacerse siempre al comienzo del script que va a usar los valores de sesión. Una vez abierta, ya permite el uso de \$SESSION.

```
1 <?php
2 session_start();
3
```

Espero sigan revisando los temas y nos encontramos la próxima clase! Consulten si hay dudas, no se queden!