# Bienvenidos



Clase 05. Introducción a PHP Validaciones – Conexión a DB

# Aplicaciones Móviles Y Cloud Computing



### Temario

04

#### Principios de PHP Validaciones – Conexión DB

- ✓ Introducción a PHP
- ✓ Validaciones con PHP
- ✓ Conexión a DB con PHP

05

#### Validaciones PHP Archivos PHP Conexión DB

- ✓ <u>Validaciones</u> con PHP
- ✓ Manejo básico archivos con PHP
- ✓ Conexión a DB con PHP

05

#### Desarrollo FullStack

- ✓ Conceptos de desarrollo FullStack
- √ Stacks de 
  programación



### Objetivos de la clase

- Aprender a realizar valicaciones con PHP
- Manejo básico de archivos con PHP
- Aprender a trabajar con PHP, en conexión con una BD



### Validaciones con PHP



### Validaciones con PHP

- ✓ isset(\$variable): valida si la variable argumento existe
- empty(\$variable): valida que la variable enviada como parámetro tenga valor
- ✓ is\_string(\$var): valida si \$var es una cadena de caracteres
- ✓ preg\_match(expr, \$var): valida si se cumple la expresión reguar expr, en \$var
- filter\_var(\$var, FILTER\_): se utiliza para validar tipos de dato especiales (email, por ejemplo). Filtra una variable, según determinado filtro.

Documentación Oficial PHP

```
(isset($ POST["nombre"]) &&
isset($_POST["email"]) &&
isset($ POST["password"])){
 $nombre=$_POST["nombre"];
 $email=$ POST["email"];
 $password=$_POST["password"];
 if(!empty($ POST["nombre"]) &&
     !empty($_POST["email"]) &&
     lempty($_POST["password"])){
     if(!is_string($nombre) || preg_match("/[0-9]/", $nombre)){
         $error="Error: nombre invalido";
     if(!is_string($email) || !filter_var($email, FILTER_VALIDATE_EMAIL)){
         $error="Error: email invalido";
     if(strlen($password)<6){
         $error="Error: password debe tener al menos 6 caracteres";
     var dump($error);
     if($error){
         header("Location:index.php?error=$error&nombre=$nombre&email=$email&password=$password");
```



# Manejo básico de archivos con PHP



### FileSystem en PHP

- √ file\_exists("nombreArchivo.json"): valida existe el archivo enviado como argumento
- ✓ file\_get\_contents("nombreArchivo.jso n"): recupera el contenido de un archivo, en formato texto
- file\_put\_contents("nombreArchivo.jso n", \$datos): escribe en el archivo al que apunta el primer argumento, los datos enviados como segundo parámetro a la función
- ✓ json\_decode y json\_encode: sirven para pasar de texto a Json, y de Json a texto, respectivamente.

**Documentacion Oficial PHP** 

```
if(file_exists("usuarios.json")){
    $json=file_get_contents("usuarios.json");
    $usuarios=json_decode($json);
}else{
    $usuarios=[];
}
```

```
$usuariosTexto=json_encode($usuarios, JSON_PRETTY_PRINT);
file_put_contents("usuarios.json", $usuariosTexto);
```



# Acceso a Base de Datos con PHP



### Conexión a Base de Datos

Existe una gran cantidad de funciones que sirven para este fin. En su mayoría comienzan con "mysqli\_". Algunos ejemplos:

- mysqli\_connect(): establece la conexión con la DB. Recibe como argumentos la dirección del server, usuario y clave, y DB
- ✓ mysqli\_connect\_errorno(): devuelve true si ocurre algún error en una instrucción ejecutada contra la DB
- ✓ mysqli\_error(): devuelve el detalle del error, si es que existe alguno.
- ✓ mysqli\_query(\$conexión, \$query): ejecuta una query en la DB indicada por \$conexión.
- msqli\_fetch\_assoc(\$query): recorre un recordset, y asigna cada fila a un array asociativo.

```
// conexión a mySQL
$conexion=mysqli_connect('localhost', 'root','','cursoupso01');
if(mysqli connect errno()){
    echo "La conexión ha DB ha fallado. ".mysqli_connect error();
}else{
    echo "Conexión a DB realizada correctamente...!!!";
mysqli query($conexion, "SET NAMES 'utf8'");
$query=mysqli_query($conexion, "SELECT * FROM usuarios");
var_dump($query);
echo "Primer print: <br>";
while($usuario=mysqli_fetch_assoc($query)){
    // var dump($usuario);
    echo $usuario['nombre']."<br>";
    echo $usuario['email']."<br>";
echo "<hr>>";
// Volver a la primera posición del resultado
mysqli_data_seek($query, 0);
```

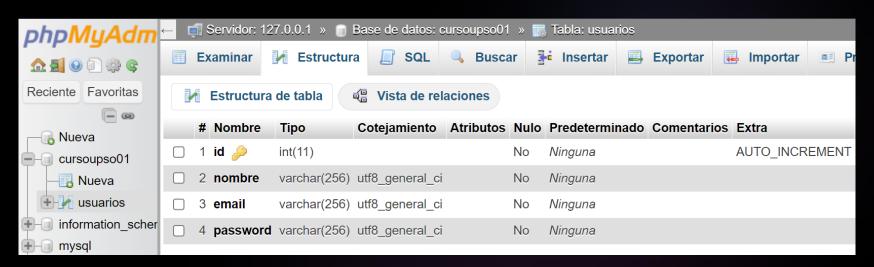


### Ejercicio:

Siguiendo el ejemplo desarrollado en clase, diseñe un login básico, que permita acceder con email y password. Considerar el mismo esquema de tabla, cuya estructura se muestra en la página siguiente.



### Ejercicio:



### Esquema tabla usuarios



### ¿Preguntas?



## Muchas gracias.

