

# LIBRERÍA JDX

Actualizado 11 abril del 2022

Esta librería de funciones está desarrollada por José de la hoz con el fin de minimizar el trabajo en el entorno de desarrollo utilizando el lenguaje PHP, esta prohibida la venta, manipulación de código, o violación de derechos de Autor.

## FUNCIONES

### fecha (Parámetros);

La función fecha(parámetro) te trae los siguientes parámetros para manejar la fecha y la hora de la siguiente forma:

Parámetros	Descripción
1	Sirve para mostrar la hora actual EJ: 10:30 PM
2	Sirve para mostrar la fecha actual completa, incluyendo la semana, día, mes y año. Ej.: lunes, 05 de abril de 2021
3	Permite mostrar la fecha solamente utilizando el siguiente formato Año: Mes: Día. Ej.: 2021-04-05
4	Permite mostrar la fecha igual que la anterior pero esta vez de manera Inversa. Ej.: 05-04-2021

Nota: Tenga en cuenta configurar la zona horaria del script, por defecto esta en : ini\_set('date.timezone','America/Bogota');

### navegador\_info (Parámetro);

La función navegador\_info(parámetro) permite mostrar algunas informaciones de los navegadores :

Parámetros	Descripción
1	Permite mostrar la dirección ip publica o privada del navegador.
2	Respuesta de tipo (Array) Permite obtener información estructurada del navegador.

## calcular\_tiempo (Parámetro\_1, Parámetro\_2 );

La función calcular\_tiempo(\$hora, \$fecha) te trae los siguientes parámetros para calcular el tiempo de alguna actividad, utilizando la hora y la fecha de la siguiente manera :

Ej : agregando las siguientes variables y estableciendo los valores de la siguiente manera :

### Fecha de Publicación

\$hora = "6:36 PM"; // debe tener este formato para no tener problemas.  
\$fecha = "2021-03-05"; // debe tener este formato para no tener problemas.

---

### Simulación de la fecha de Actual

\$hora = "6:36 PM"; // debe tener este formato para no tener problemas.  
\$fecha = "2021-04-05"; // debe tener este formato para no tener problemas.

---

\$calcular\_tiempo = calcular\_tiempo(\$hora, \$fecha);

\$fpub = "El usuario administrador público un artículo \$calcular\_tiempo";

Echo \$fpub; // El usuario administrador público un artículo hace un día.

Las siguientes descripciones varían dependiendo el tiempo de publicación y extrayendo la hora y fecha.

Descripciones
Hace un momento
Hace 3 minutos // Ej
Hace un día
Hace un mes
Hace un año

## reemplazar (Parámetros\_1, Parámetros\_2, Parámetros\_3);

La función reemplazar(\$cadena, \$buscarvalor, \$reemplazar) te trae los siguientes parámetros para reemplazar un valor de una cadena de la siguiente forma :

\$cadena = "juan esta en el parque";

\$buscar = "parque";

\$reemplazar = "sofa";

Echo reemplazar(\$cadena, \$buscar, \$reemplazar);

// El resultado será : juan esta en el Sofa.

## randalfa (Parámetros\_1, Parámetros\_2);

La función randalfa(\$tipo, \$rango) te trae los siguientes parámetros para generar aleatoriamente números con letras de la siguiente forma :

Parámetros	Descripción
1,10	Genera una mezcla de letras en mayúsculas y números de un rango de 10. Resultado: YQdPZv5ynO
2,10	Genera una mezcla de letras en minúsculas y números de un rango de 10. Resultado : hw61t85nec

## randnum (Parámetro);

La función randnum(\$rango) te trae los siguientes parámetros para generar aleatoriamente números de la siguiente forma :

Parámetros	Descripción
10	Genera aleatoriamente números de un rango de 10. Resultado: 2436864944
El rango máximo para generar números aleatoriamente es de 50.	

## base64\_imagen (Parámetro);

La función base64\_imagen() te permite obtener la codificación en base64 de una imagen para poder abrirla desde el navegador sin la necesidad de tener la imagen descargada o también se puede utilizar para otras situaciones :

Ej : base64\_imagen("foto.png"); // El resultado será el siguiente :

Has click para abrir la codificación de la imagen en el navegador

 <- Imagen Facebook

Codigo codificado de la Imagen

```
=====

AB/pwA+AAAAKlBMVEWXP8j5+vr8/P35+frm6fD4+PmBlb1DYJw7WZhbdKINa
al/XJrK0uN6jrmkQ+p1AAAATUIEQVR4AWNgVIICAQYhGFORAUlruyopQZiq6R
2ToEy1jo4iKFORoxvG1OhwUolxu4EETBTG7ACCHgSzFMG8hKSWbKYQjKnIwAh
jCgAAUH8ojs+8AMcAAAAASUVORK5CYII=
```

Si se copia este código y se introduce en la barra de direcciones del navegador podemos obtener la imagen de Facebook.

## antihack(Parámetro);

La función antihack() te permite filtrar los valores que entran en la función permitiendo la seguridad en las base de datos antes posibles ataques SQL Injection.

Ej :

\$valor = "Juan ' or 1=1' ";

Con la función daría como resultado => Juan \' or 1=1 \'

La cual permite que no se inserte los string ' directamente sino que haya al momento de entrar un escape y no haya problemas de errores en la DB.

Sin la función daría como resultado => Juan ' or 1=1 '

Nos saldría entonces un error de sintaxis peligroso para la seguridad de la DB.

### **eliminar\_archivo** (Parámetro);

La función eliminar\_archivo(\$ruta\_archivo) te permite eliminar cualquier archivo introduciendo la ruta de destino.

La respuesta si se elimino es igual a 1 de lo contrario se mostrará el error.

### **eliminar\_carpeta** (Parámetro);

La función eliminar\_carpeta(\$ruta\_carpeta, \$ver\_historial) te permite eliminar una carpeta utilizando la ruta de donde se encuentra ubicada la carpeta.

La respuesta si se elimino es igual a 1 de lo contrario se mostrara el error.

Nota : si queremos ver todo lo que se elimino podemos poner un 1 en \$ver\_historial y sino borramos la variable.

### **ver\_contenido** (Parámetro);

La función ver\_contenido(\$ruta\_archivo) te permite ver el contenido que contiene un archivo introduciendo la ruta de destino.

### **copiar\_archivo** (Parámetro\_1, Parámetro\_2);

La función copiar\_archivo(\$ruta\_archivo, \$ruta\_destino) te permite copiar un archivo a un destino introduciendo la ruta del archivo a copiar y ruta destino donde se desea que se realice la copia.

La respuesta si se realizo la copia exitosamente es igual a 1 de lo contrario se mostrara el error.

### **crear\_archivo (Parámetro\_1, Parámetro\_2);**

La función crear\_archivo(\$nombre\_archivo, \$contenido) te permite crear y agregar en un archivo contenido generado por uno mismo por ejemplo :

```
$contenido = "<?php echo 'Hola Mundo este es una prueba de la función  
crear_Archivo() ' ; ?> ";
```

```
$nombre_archivo = "prueba.php";
```

```
echo crear_archivo($nombre_archivo, $contenido);
```

La respuesta si se creo exitosamente es igual a 1 de lo contrario se mostrará el error.

### **renombrar (Parámetro\_1, Parámetro\_2);**

La función renombrar(\$nombre\_antiguo, \$nombre\_nuevo) te permite renombrar el nombre de un archivo por ejemplo :

```
$nombre_antiguo = "prueba.php";
```

```
$nombre_nuevo = "prueba-test.php";
```

```
Echo renombrar($nombre_antiguo, $nombre_nuevo);
```

```
// Si la respuesta es 1 se renombro exitosamente de lo contrario se  
mostrara el error.
```

### **crear\_carpeta (Parámetro\_1, Parámetro\_2);**

La función crear\_carpeta(\$nombre\_de\_carpeta, \$ruta) te permite crear una carpeta introduciendo el nombre y la ruta donde se creara.

Si desea crearla en la ubicación donde esta en ruta puede poner 0, crear\_carpeta("prueba", 0).

La respuesta si se creo es igual a 1 de lo contrario se mostrará el error y si la ruta no existe se creara en la ubicación donde esta.

### **existe\_carpeta (Parámetro);**

La función existe\_carpeta(\$nombre\_de\_carpeta o \$Ruta) te permite verificar si la carpeta existe introduciendo la ruta o la ubicación donde esta.

La respuesta si existe es igual a 1 de lo contrario se mostrará el error.

### **existe\_archivo (Parámetro);**

La función existe\_archivo(\$nombre\_del\_archivo o \$Ruta) te permite verificar si existe un archivo introduciendo la ruta o la ubicación donde esta.

La respuesta si existe es igual a 1 de lo contrario se mostrará el error.

### **comprimir\_carpeta (Parámetro\_1, Parámetro\_2, Parámetro\_3);**

La función comprimir\_carpeta(\$carpeta, \$nombre\_zip, \$rutadestino) te permite comprimir una carpeta con todo el contenido que pueda tener utilizando la encriptación ZIP.

Nota: si usted quiere crear el archivo comprimido en la ubicación donde esta entonces en ruta de destino elimine la variable ruta destino.

La respuesta si la compresión fue exitosa es igual a 1 de lo contrario se mostrará el error.

### **descomprimir\_archivo (Parámetro\_1, Parámetro\_2);**

La función descdomprimir\_archivos(\$nombre\_zip, \$rutadestino) te permite descomprimir archivos en formato zip.

Nota: si usted quiere descomprimir el archivo en la ubicación donde esta entonces elimine la variable rutadestino.

La estructura es de la siguiente manera:

```
$nombre_zip = "fotos_2021";
```

Echo descomprimir\_archivo(\$nombre\_zip);  
//Me traerá como respuesta 1 si se descomprimió correctamente de lo contrario se mostrara el error.

### **extraer\_extencion (Parámetro);**

La función extraer\_extencion(\$nombre\_archivo) te permite extraer la extencion de un archivo especificando el nombre.

Ej.

```
echo extraer_extencion("index.php"); // La extensión será PHP
```

Si la respuesta no nos trae la extensión del archivo se mostrará el error del problema.

### **nombre\_base (Parámetro\_1, Parámetro\_2);**

La función nombre\_base(\$nombre\_archivo, \$extraer) es casi parecida a la función extraer\_extencion sino que en este caso te traerá el nombre del archivo con sin la extencion.

Ej.

```
echo nombre_base ("file/accounts/hack.txt", 1); // El valor que nos mostrara será index sin la extencion de lo contrario si es 2 es con el nombre del archivo mas la extencion.
```

Si la respuesta no nos trae el valor correspondiente se mostrará el error del problema.

### **info\_servidor (Parámetro);**

La función info\_servidor(Parametro) nos permite obtener información limitada del servidor.

Parámetros	Descripción
host	Obtener el host del servidor
protocolo	Obtener el protocolo del servidor



tipo_protocolo	Obtener el tipo de protocolo del servidor
mi_url	Obtener el enlace de la pagina web con el protocolo y host.

### **codificar\_json (Parámetro);**

La función `codificar_json($array)` nos permite codificar valores que estan dentro de un array a formato json de la siguiente manera:

```
$array = array("usuario"=>"Juan", "clave"=>"1234");
echo codificar_json($array);

// La codificación será {"usuario":"Juan","clave":"1234"}
```

La cual nos servirá para la utilización de una API.

### **decodificar\_json (Parámetro);**

La función `decodificar_json($JSON)` nos permite decodificar valores que estan codificado en formato json de la siguiente manera:

```
$JSON = ' {"usuario":"Juan","clave":"1234"} ';
echo decodificar_json($JSON)["usuario"]; // Me mostrara Juan

// La decodificación será {"usuario":"Juan","clave":"1234"}
```

Si la respuesta no nos trae el valor a solicitar se mostrará el error del problema.

### **pl\_mayus (Parámetro);**

La función `pl_mayus($cadena)` nos permite convertir la primera letra de una cadena de texto en mayusculas.

### **pl\_min (Parámetro);**

La función pl\_min(\$cadena) nos permite convertir la primera letra de una cadena de texto en minúsculas.

### **con\_mayus (Parámetro);**

La función con\_mayus(\$cadena) nos permite convertir una cadena de texto en mayúsculas.

### **con\_minus(Parámetro);**

La función con\_min(\$cadena) nos permite convertir una cadena de texto en minúsculas.

### **extraer\_edad(Parámetro);**

La función extraer\_edad(\$fecha\_nacimiento) nos permite obtener la edad la persona introduciendo solamente la fecha de nacimiento de la siguiente manera :

El formato de fecha debe ser : Año – Mes - Día de lo contrario mostrara un error.

---

```
echo extraer_edad("2000-12-05"); // La persona tiene 21 años
```

### **seguridad\_clave(Parámetro\_1, Parámetro\_2);**

La función seguridad\_clave(\$clave, \$hash) nos permite encriptar nuestras contraseñas con un cifrado seguro de la siguiente manera :

#### **Para encriptar**

```
$clave = "jdx6577"; // Variable definida con la clave que le vamos a poner al encriptado.
```

---

```
echo seguridad_clave($clave);
```

```
//Nos traerá de manera aleatoria el hash encriptado de esa contraseña =>  
$2y$10$LihI/vUo5ph9Ww6mAtVY7OZkGm.Jl6w8rhGXhy9T/9ksVS9nbibN.
```

## Para descriptar

Desafortunadamente es imposible descriptar un hash que esta codificado y esto se debe a que se desarrolló una manera de encriptar las contraseñas de una sola via.

## Para verificación de claves

```
$hash =  
"$2y$10$Lihi/vUo5ph9Ww6mAtVY7OZkGm.Jl6w8rhGXhy9T/9ksVS9nbibN."  
";
```

```
// Variable definida con el hash que vamos a verificar.
```

```
$clave = "jdx6577"; // Variable definida con la clave que vamos a verificar.
```

```
-----  
echo seguridad_clave($clave, $hash);
```

```
//Si la clave es correcta con el hash que se encripto nos traerá el valor 1,  
pero si la clave o el hash son incorrecto nos mostrara el valor 0;
```

## extraer\_edad(Parámetro\_1);

La función extraer\_edad(\$fecha) nos permite obtener la edad real de un usuario utilizando la fecha de la siguiente manera:

\$fecha = "2000-12-05"; El formato debe ser año – dia – mes.

```
echo extraer_edad($fecha); // Nos traerá el valor de 20 lo que significa  
que la persona en el año 2021 antes del mes de diciembre tenia 20 Años.
```

## verificar\_coincidencia(Parámetro\_1, Parámetro\_2);

La función verificar\_coincidencia(\$cadena\_texto, \$array) nos permite verificar a través de un array si existe coincidencia en una porción o cadena de texto utilizando los valores que estan establecidos en el array, funciona de la siguiente manera :

```
$cadena = "Esta es una prueba de juan utilizando la dicha Funcion";
```

```
$array = array("maria", "juan"); //Palabras a buscar
```

`echo verificar_coincidencias($cadena, $array); // Nos traerá el valor de 1 si se encontró algunos de los valores establecidos en el array dentro de la cadena de texto y valor nulo si NO se encontró ningún valor del array en la cadena de texto.`

Si la respuesta no nos trae el valor 1 ni tampoco el valor nulo, se mostrará el error del problema.

### **verificar\_archivo\_subir(Parámetro\_1, Parámetro\_2);**

La función `verificar_archivo_subir($nombre_file, $array)` nos permite verificar a través de un array si el archivo a subir cumple el formato deseado utilizando los valores que están establecidos en el array, funciona de la siguiente manera :

```
$array = array("jpg","png", "gif", "webp");  
$file = "image"; //Esto viene del input file con name -> image
```

`Echo verificar_archivo_subir($file, $array); // Si el archivo a subir cumple algunos de los formato del array nos dará como respuesta el valor 1 de lo contrario será null o vacío.`

Nota : Esta función es experimental ya que protege de ataques de FileUpload.

### **verificar\_fecha (Parámetro\_1, Parámetro\_2);**

La función `verificar_fecha($fecha, $formato)` nos permite verificar la estructura de una fecha u hora utilizando el formato de la siguiente manera:

```
$formato = "Y-m-d";  
$fecha =" 2000-12-01";
```

`echo verificar_fecha($fecha, $formato); // Traera como resultado 1 que significa que el formato y la fecha introducida son correctos de lo contrario si no es valido la fecha con el formato nos traerá como valor nulo, vacío.`

### **verificar\_caducidad** (Parámetro\_1, Parámetro\_2);

La función verificar\_caducidad(\$hora, \$fecha) nos permite calcular el tiempo con la hora y fecha con el fin de utilizarlo como mini temporizador en el entorno de programación, funciona de la siguiente manera :

```
$hora = "12:25:00"; // El formato de la hora debe ser de 24 horas.
```

```
$fecha = "21-05-2021"; // El formato de la fecha debe ser día - mes - año.
```

```
echo verificar_caducidad($hora, $fecha); // Si la fecha real es la misma de la fecha que está establecida y la hora es menor a la hora que también está establecida nos traerá el valor de 0 lo que significa que no se ha concretado la hora y fecha que está establecida de lo contrario si la hora es mayor a la que está establecida nos traerá el valor de 1.
```

### **descuento** (Parámetro\_1, Parámetro\_2);

La función descuento(\$valor, \$descuento) nos permite hacer descuentos por porcentajes de la siguiente manera :

```
$base = "100500"; // El numero debe escribirse sin puntos.
```

```
$descuento = 10; // El numero de descuento debe escribirse sin los porcentajes.
```

```
echo descuento($base, $descuento); // Nos traerá el valor de 90450
```

### **cformato** (Parámetro\_1, Parámetro\_2);

La función cformato(\$valor, \$descuento) nos permite con dar formato de miles a un numero de la siguiente manera :

```
$valor = "10000";
```

```
echo cformato($valor,1); // Nos traerá el valor de 10.000
```

Parámetros	Descripción
1	Nos dará el fomato de miles con puntos.
2	Nos dará el fomato de miles con comas.

## **curl\_view (Parámetro\_1, Parámetro\_2);**

La función curl\_view(\$url, \$array) nos permite acceder a una pagina web y si deseamos enviar parametros especificos, funciona de la siguiente manera :

```
$url = "https://j-doxx.000webhostapp.com/login";
```

```
$array = array("usuario"=>"admin", "clave"=>1234);
```

```
//Echo curl_view($url); // Accedera la pagina sin iniciar sesión.
```

- POR METODO GET

```
//Echo curl_view($url, $array); // Accedera la pagina e iniciara sesión.
```

- POR METODO POST

## **url\_amigable (Parámetro\_1);**

La función url\_amigable(\$nombre\_contenido) nos permite convertir el nombre de un contenido en url amigable, funciona de la siguiente manera :

```
$nombre_contenido = "Curso Mysql javascript 2022";
```

```
Echo url_amigable($nombre_contenido);
```

```
// El valor a entregar es -> curso-mysql-javascript-2022
```

## **calcular\_tamano\_archivo (Parámetro\_1);**

La función calcular\_tamano\_archivo(\$archivo) nos permite obtener el tamaño del archivo en formato conocible, funciona de la siguiente manera :

```
$archivo = "video.mp4";
```

```
Echo calcular_tamano_archivo($archivo);
```

```
//El valor de entrega será como ej : 10.33 MB si el video en realidad pesa ese numero, de lo contrario se mostrara el error.
```

### **obtener\_query\_param (Parámetro\_1, Parámetro\_2);**

La función obtener\_query\_param(\$url, \$param) nos permite obtener un valor que se encuentra en una petición GET, funciona de la siguiente manera :

```
$url = "http://j-  
doxx.000webhostapp.com/login.php?user=admin&pass=1234";  
$param = "user";
```

```
Echo obtener_query_param($url, $param);
```

```
//El valor de entrega será -> admin, de lo contrario se mostrara el error.
```

### **obtener\_unix\_temporizador (Parámetro\_1, Parámetro\_2);**

La función obtener\_unix\_temporizador(\$sigla, \$valor) nos permite obtener en formato UNIX el tiempo según el valor establecido para utilizarlo como temporizador mas adelante, funciona de la siguiente manera :

Las siglas son las siguientes (d, h, m, s) y funciona asi.

Por ej : quiero establecer un temporizador de 10 minutos, se realizaría de la siguiente manera.

```
Echo calcular_unix_time("m",10);
```

El valor que nos entregara según la hora y fecha donde vive será algo parecido asi Ejemplo : 1649676103.

Ya con el valor UNIX (1649676103) podemos utilizarlo en la siguiente función.

## calcular\_unix\_time (Parámetro\_1);

La función obtener\_unix\_temporizador(\$valor\_unix) nos permite obtener el tiempo restante establecido calculándolo con la fecha y hora de donde vive, funciona de la siguiente manera :

```
$valor_unix = 1649676103;
```

```
Echo calcular_unix_time($valor_unix);
```

```
//El valor de entrega será -> D-00 00:09:48, de lo contrario se mostrara el error.
```

Como puede ver es un reloj calculando el tiempo restante hasta que se acabe el tiempo de 10 minutos, cuando se acabe el tiempo nos dará como respuesta el numero 2.

## INSTALACION

La incorporación de la herramienta es sencilla y es incluyendo nuestro archivo en el código PHP de la siguiente manera:

```
<?php
```

```
Include "LIB-JDX.php";
```

```
?>
```

Después de hacer esto ya podemos ejecutar las funciones preestablecidas.

## MANEJO DE ERRORES

Todas las funciones prediseñadas devuelven siempre un valor la cual se describe si hay un error o si algo esta mal en los parámetros, para ejecutar las funciones se hace de la siguiente forma:

### Simulacion de la función crear\_carpeta()

La función crear\_carpeta() simulara que el directorio juan se creo exitosamente de la siguiente manera :

```
$carpeta = "Juan";
```

```
$ruta = 0; // 0 porque queremos creala en la ubicación donde estamos
```



```
If (crear_carpeta($carpeta) == 1){  
Echo "La carpeta $carpeta se creo Exitosamente ";  
}else{  
Echo "La carpeta $carpeta NO no se pudo crear ";  
}
```

Si ha habido un error, el error no te lo mostrara por eso es necesario verlo de la siguiente manera :

```
echo crear_carpeta($carpeta);
```

Aquí nos mostrara el error por ejemplo podría ser los siguientes errores :

- Error ya Existe la carpeta !
- Error faltan Parametros ! // Si se ejecuta la funcion sin ningún parámetro o si hacen faltan parámetros.

## ACTUALIZACION

Nuestra herramienta

## PROBLEMAS EN LA HERRAMIENTA

Si llega a presentar errores la librería no dude en comunicarse con el desarrollador a través de whatsapp +573117304319 o a través de correo electrónico [josealfonzod@gmail.com](mailto:josealfonzod@gmail.com).

Nuestra página Web -> [Has click aquí](#)).



**COPYRIGHT – 2022**

Todos los derechos reservados