

# LIBRERÍA JDX VERSION 1.0

Actualizado 2 Junio del 2021

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

### 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	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
randalfa(1,10);	Genera una mezcla de letras en mayúsculas y números de un rango de 10. Resultado: YQdPZv5ynO
randalfa(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
randnum(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+AAAAKIBMVEWXp8j5+vr8/P35+frm6fD4+PmBlb1DYJw7WZhbdKlNa
aI/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) 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.

### **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.

### **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 escriba un cero.

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

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

La función comprimir\_archivos(\$nombre\_zip, \$array, \$rutadestino) te permite comprimir archivos especificando el nombre de cada uno agregándolos en un array, la encriptación es igual que la de Arriba.

Nota: si usted quiere crear el archivo comprimido en la ubicación donde esta entonces en ruta de destino escriba un cero.

La estructura es de la siguiente manera:

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

```
$rutadestino = 0;
```

```
$array = array("foto_v1.png", "foto_v2.png", "foto_v3.png")
```

```
Echo comprimir_archivos($nombre_zip, $array, $rutadestino);
```

```
//Me traerá como respuesta 1 y me generara el archivo comprimido.
```

```
El archivo si lo buscamos aparecerá así fotos_2021.zip
```

La respuesta si la compresión fue exitosa es igual a 1 de lo contrario se mostrará 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);**

La función `nombre_base($nombre_archivo)` es casi parecida a la función `extraer_extencion` sino que en este caso te traerá solamente el nombre.

Ej.

```
echo nombre_base ("index.php"); // El nombre será index sin la extencion
```

Si la respuesta no nos trae el nombre solo del archivo 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 `dedificar_json($array)` 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, Parámetro\_3);**

La función seguridad\_clave(\$hash, \$clave, \$modo) 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(0, $clave, 1);
```

```
//Nos traerá el hash encriptado de esa contraseña =>
```

```
3e39d6a7ae392171b5b37f59a784321aaade95e52801eb09ce3fd047
```

Para desencriptar

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

### Para verificación de claves

```
$hash =  
"3e39d6a7ae392171b5b37f59a784321aaade95e52801eb09ce3fd047";  
  
// Variable definida con el hash que vamos a verificar.  
  
$clave = "jdx6577"; // Variable definida con la clave que vamos a verificar.
```

```
-----  
  
echo seguridad_clave($hash, $clave, 2);  
  
//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;
```

Nota : Para encriptar una clave en el parámetro hash escriba un 0  
quedaria asi :

```
// echo seguridad_clave(0, $clave, 1);
```



### 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\_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, vacio.
```

### **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.

## INSTALACION

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

```
<?php
```

```
Include "JDX_1.0.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:

### Simulación 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 se puede actualizar de la siguiente manera :

Se puede crear un archivo PHP con el nombre actualizar.php con el siguiente contenido :

```
<?php
```

```
Include "JDX_1.0.php"; // Nombre de librería con la versión que dispone.
```

```
Echo actualizar_version();
```

```
?>
```

O desde donde se esta trabajando

```
<?php
```

```
Include "JDX_1.0.php";
```

```
Echo actualizar_version();
```

```
Exit();
```

<----- ABAJO PUEDE HABER CODIGO PHP HTML ETC ----->

```
?>
```

---

Al momento que se ejecute la funcion esta le mostrara si la versión que usted dispone esta actualizada y sino NO lo esta le mostrara las instrucciones de descarga y actualizacion.

## PROBLEMAS EN LA HERRAMIENTA

Posiblemente cuando se ejecute la funcion actualizar\_version() si no nos muestra una respuesta concreta es necesario que se comuniquen con el creador a traves de whatsapp +573117304319 o a través de correo electrónico [josealfonzod@gmail.com](mailto:josealfonzod@gmail.com).

Esto se debe a que la pagina web, en la API donde se conecta se ha trasladado a una nueva dirección. Nuestra página Web -> [Has click aquí](#)).

**COPYRIGHT – 2021**

Todos los derechos reservados por el creador José de la Hoz