



**Clase 14**

# Análisis de Sistemas

**Materia:**  
Programación Web II

**Docente contenidista:** ROLDÁN, Hernán

**Revisión:** Coordinación

# Contenido

PHP Manejo de fecha y hora.....	3
Introducción a la Clase DateTime .....	3
Obtener la fecha y hora actual .....	3
Calcular intervalos de tiempo .....	4
Formatear fechas .....	6
Trabajar con zonas horarias.....	7
Ejemplo práctico: Cyber Monday .....	8
Bibliografía .....	11
Para ampliar la información .....	11

# Clase 14



¡Te damos la bienvenida a la materia  
**Programación Web II!**

**En esta clase vamos a ver los siguientes temas:**

- Obtener la fecha y hora actual.
- Calcular intervalos de tiempo.
- Formatear fechas.
- Trabajar con zonas horarias.
- Ejemplo práctico: Cyber Monday.

# PHP Manejo de fecha y hora

## Introducción a la Clase DateTime

PHP proporciona una variedad de funciones y formatos para trabajar con fechas y horas en aplicaciones web. El manejo adecuado de fechas y horas es esencial para diversas tareas, como mostrar la fecha actual, calcular intervalos de tiempo, formatear fechas para mostrarlas a los usuarios y trabajar con zonas horarias.

En PHP, las fechas y horas se representan mediante objetos de la clase DateTime. Esta clase nos permite realizar diversas operaciones, como obtener la fecha y hora actual, sumar o restar intervalos de tiempo, formatear fechas y mucho más.

## Obtener la fecha y hora actual

Para obtener la fecha y hora actual en PHP, podemos usar la función `date()`. Esta función acepta un parámetro opcional para definir el formato en el que queremos mostrar la fecha y hora. Si no se proporciona ningún parámetro, la función mostrará la fecha y hora actual en formato predeterminado.

Ejemplo de uso de la función `date()`:

```
<?php

    $fechaActual = date('Y-m-d H:i:s'); // Formato 'Año-Mes-
    Día Hora:Minutos:Segundos'
    echo "Fecha y Hora Actual: " . $fechaActual;

?>
```

# Calcular intervalos de tiempo

PHP nos permite realizar operaciones con fechas y horas, lo que nos permite calcular intervalos de tiempo. Podemos usar la función **strtotime()** para convertir una cadena con formato de fecha en un valor de tiempo Unix, que es el número de segundos desde el 1 de enero de 1970. Luego, podemos realizar cálculos matemáticos para obtener la diferencia entre dos fechas.

Ejemplo de uso de la función strtotime():

```
<?php

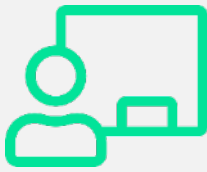
$fechaInicio = strtotime('2023-01-01');
$fechaFin = strtotime('2023-12-31');

var_dump($fechaInicio); // 1672531200 segundos
var_dump($fechaFin); // 1703980800 segundos

$intervalo = $fechaFin - $fechaInicio; // 1703980800 -
1672531200 = 31449600 segundos
$numeroDias = $intervalo / (60 * 60 * 24); // Convertir
segundos a días (31449600 / 86400 = 364)

echo "Número de días entre el 1 de enero y el 31 de
diciembre de 2023: " . $numeroDias . " días";

?>
```



### Explicación del código:

Lo primero que hacemos es obtener la fecha y hora de inicio y fin:

- Utilizamos la función **strptime()** para convertir las cadenas de fecha en valores de tiempo Unix.
- Estos valores representan las fechas de inicio y fin del intervalo que queremos calcular.

Calcular el intervalo de tiempo:

- Restamos el valor de tiempo Unix de la fecha de inicio del valor de tiempo Unix de la fecha de fin.
- Obtenemos el intervalo de tiempo en segundos.

Convertir el intervalo de tiempo a días:

- Dividimos el intervalo en segundos entre el número de segundos en un día (60 segundos \* 60 minutos \* 24 horas).
- Obtenemos el número total de días entre las dos fechas.

### Mostrar el resultado:

- Mostramos el resultado en pantalla utilizando la función echo.

Este proceso nos permite calcular la diferencia entre dos fechas y obtener el intervalo de tiempo en días. Puede aplicarse en diversas situaciones, como calcular la duración de un proyecto, el tiempo entre dos eventos o la antigüedad de un registro, entre otros.

# Formatear fechas

En ocasiones, queremos mostrar fechas en un formato específico. Para ello, podemos utilizar el método **format()** de la clase DateTime. Este método acepta un parámetro que define el formato deseado.

Ejemplo de uso de la función format():

```
<?php

$fecha = new DateTime('2023-07-19');

echo "Fecha en formato d-m-Y: " . $fecha->format('d-m-Y') . "<br>";
// Día-Mes-Año (19-07-2023)
echo "Fecha en formato Y-m-d H:i:s: " . $fecha->format('Y-m-d
H:i:s') . "<br>"; // Año-Mes-Día Hora:Minutos:Segundos (2023-07-19
00:00:00)
echo "Fecha en formato l, d F Y: " . $fecha->format('l, d F Y') .
"<br>"; // Día de la semana, Día Mes Año (miércoles, 19 julio 2023)

?>
```

# Trabajar con zonas horarias

Cuando trabajamos con fechas y horas, es esencial considerar las zonas horarias. PHP nos permite establecer una zona horaria para nuestras fechas usando la función **date\_default\_timezone\_set()**.

Ejemplo de uso de la función `date_default_timezone_set()`:

```
<?php

    date_default_timezone_set('America/Argentina/Buenos_Aires');

    $fecha = new DateTime();
    echo "Fecha y Hora en Buenos Aires: " . $fecha-
>format('Y-m-d H:i:s');

?>
```

En este ejemplo, establecemos la zona horaria a "America/Argentina/Buenos\_Aires". De esta manera, todas las operaciones de fecha y hora que realicemos en el script se ajustarán a la zona horaria de Buenos Aires.

PHP Date and Time es una parte fundamental en el desarrollo de aplicaciones web, ya que nos permite trabajar con fechas, horas y zonas horarias de manera efectiva. Con las funciones y métodos proporcionados por PHP, podemos obtener la fecha y hora actual, calcular intervalos de tiempo, formatear fechas para mostrarlas a los usuarios y trabajar con diferentes zonas horarias. Estas habilidades son esenciales para crear aplicaciones web dinámicas y con una presentación adecuada de las fechas y horas.



# Ejemplo práctico: Cyber Monday

Supongamos que estamos desarrollando una aplicación de comercio electrónico para nuestra tienda online y queremos ofrecer a nuestros clientes descuentos especiales durante el evento "Cyber Monday". Sin embargo, deseamos que estas ofertas solo sean válidas durante un período de 24 horas, comenzando desde el inicio del día lunes en que se celebra el evento.

Para implementar esta funcionalidad con PHP, definiremos una función llamada **esOfertaActiva(\$fecha\_inicio, \$duracion\_horas)** que tomará como parámetros la fecha y hora de inicio de la oferta y la duración en horas que queremos que esté activa.

Dentro de la función, utilizaremos la función **time()** de PHP para obtener el timestamp actual, que representa la fecha y hora actuales en segundos desde el epoch (1 de enero de 1970). Luego, convertiremos la fecha de inicio de la oferta a un timestamp utilizando la función **strtotime()**, que nos permitirá comparar las fechas en formato numérico.

Calcularemos la fecha límite de la oferta sumando la duración en horas al timestamp de inicio. Después, compararemos si el timestamp actual se encuentra dentro del rango entre el inicio y la fecha límite de la oferta para determinar si la oferta está activa en el momento actual.

Finalmente, en el código principal, definiremos la fecha de inicio del Cyber Monday y la duración de 24 horas para la oferta. Luego, llamaremos a la función **esOfertaActiva()** con estos parámetros y mostraremos un mensaje adecuado según si la oferta está activa o no.

El resultado indicará si la oferta Cyber Monday está activa en el momento en que se ejecuta el código, permitiéndonos ofrecer descuentos especiales a nuestros clientes durante las próximas 24 horas a partir del inicio del día lunes.

Nuestro código quedaría así:

```
<?php

// Supongamos que estamos desarrollando una aplicación de
ecommerce y queremos lanzar ofertas Cyber Monday que solo
puedan ejecutarse dentro de las 24 hs.

// Función para verificar si la oferta está activa
function esOfertaActiva($fecha_inicio, $duracion_horas) {
```

```

    $fecha_actual = time();
    $fecha_inicio_ts = strtotime($fecha_inicio);
    $fecha_fin_ts = $fecha_inicio_ts + ($duracion_horas *
3600); // Convertir horas a segundos

    return ($fecha_actual >= $fecha_inicio_ts &&
$fecha_actual <= $fecha_fin_ts);
}

// Datos de la oferta Cyber Monday
$fecha_inicio_oferta = "2023-07-21 00:00:00"; // La
oferta comienza el 1 de diciembre de 2023 a las 00:00:00
$duracion_oferta_horas = 24; // La oferta estará activa
durante 24 horas (hasta las 23:59:59 del 1 de diciembre de
2023)

// Verificar si la oferta está activa
if (esOfertaActiva($fecha_inicio_oferta,
$duracion_oferta_horas)) {
    echo ";Oferta Cyber Monday activa!\n";
    echo ";Aprovecha nuestros descuentos especiales
durante las próximas 24 horas!\n";
} else {
    echo "La oferta Cyber Monday aún no ha comenzado o ya
ha finalizado.\n";
    echo ";Vuelve el próximo Cyber Monday para aprovechar
nuestros descuentos!\n";
}
?>

```

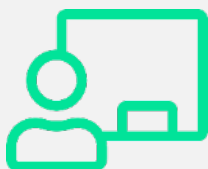
*Recordá que el manejo de fechas y horas es una parte esencial del desarrollo web, y dominar estas habilidades te permitirá crear aplicaciones más robustas y precisas.*

*Para más información sobre el manejo de fecha y hora visité la documentación oficial de PHP, en la sección específica de las clases Date y Time.*



Hemos llegado así al final de esta clase en la que vimos:

- Obtener la fecha y hora actual.
- Calcular intervalos de tiempo.
- Formatear fechas.
- Trabajar con zonas horarias.
- Ejemplo práctico: Cyber Monday.



Te esperamos en la **clase en vivo** de esta semana.  
No olvides realizar el **desafío semanal**.

**¡Hasta la próxima clase!**

# Bibliografía

[Documentación Oficial de PHP](#)

[W3Schools: PHP Tutorial](#)

Cabezas Granado, Luis. González Lozano, F. (2018) Desarrollo web con PHP y MySQL. Anaya Multimedia.

Heurtel, O. (2016) Desarrolle un sitio web dinámico e interactivo. Eni Ediciones.

Welling, Luke Thompson, L. (2017) Desarrollo Web con PHP y MySQL. Quinta Edición (2017, Editorial Anaya).

## Para ampliar la información

[PHP: Fecha/Hora](#)

[PHP: Date and Time](#)

[PHP: Date and Time Functions](#)

[PHP: The Right Way](#)

[Todo sobre PHP](#)

[PHP Programming Language Tutorial - Full Course](#)

[PHP Full Course for non-haters](#)

[CURSO de php desde cero](#)

[MDN: Introducción al lado Servidor](#)

[Guía de HTML](#)