

Excepciones

<https://www.php.net/manual/es/language.exceptions.php>

<https://www.php.net/manual/es/pdo.setattribute.php>

<https://www.php.net/manual/es/class.pdo.php>

<https://www.php.net/manual/es/class.exception.php>

Excepciones

- A partir de la versión 5 se introdujo en PHP un modelo de excepciones similar al existente en otros lenguajes de programación.
- PHP ofrece una clase base Exception para utilizar como manejador de excepciones.
- Para lanzar una excepción no es necesario indicar ningún parámetro, aunque de forma opcional se puede pasar un mensaje de error y también un código de error.
- Entre los métodos que puedes usar con los objetos de la clase Exception están:
 - **getMessage**. Devuelve el mensaje, en caso de que se haya puesto alguno.
 - **getCode**. Devuelve el código de error si existe.

Excepciones

- El código susceptible de producir algún error se introduce en un bloque **try**.
- Cuando se produce algún error, se lanza una excepción utilizando la instrucción **throw**.
- Después del bloque try debe haber como mínimo un bloque **catch** encargado de procesar el error o un bloque **finally**.
- Si una vez acabado el bloque try no se ha lanzado ninguna excepción, se continúa con la ejecución en la línea siguiente al bloque o bloques catch (si existe un bloque finally, este se ejecutará se haya producido o no alguna excepción).

Excepciones

- Por ejemplo, para lanzar una excepción cuando se produce una división por cero podrías hacer:

```
try {  
    if ($divisor == 0)  
        throw new Exception("División por cero.");  
  
    $resultado = $dividendo / $divisor;  
}  
catch (Exception $e) {  
    echo "Se ha producido el siguiente error: ".$e->getMessage();  
}
```

Si se produce la excepción, se mostrará el mensaje: 'Se ha producido el siguiente error: división por cero.'

Excepciones

- Se pueden usar varios bloques catch para atrapar diferentes clases de excepciones.
- Cuando una excepción es lanzada, la ejecución del código se detiene, y PHP intentará encontrar el primer bloque catch coincidente.
- Si una excepción no es capturada, se emitirá un Error Fatal de PHP con un mensaje "Uncaught Exception ..." ("Excepción No Capturada"), a menos que se haya definido un manejador con `set_exception_handler()`.
- Se puede utilizar un bloque finally después o en lugar de los bloques catch.
- El código de dentro del bloque finally siempre se ejecutará después de los bloques try y catch, independientemente de que se haya lanzado una excepción o no, y antes de que la ejecución normal continúe.