

# **Control de Errores**

Antonio Espín Herranz

# Tipos de Errores

- **Error de análisis:** Es un problema con la sintaxis, por ejemplo, falta un ;
- **Error Fatal:** Llamar a una función que no se ha definido.
- **Advertencia:** Algo va mal pero el intérprete puede continuar. Por ejemplo, número erróneo de argumentos en una función.
- **Aviso:** Imprimir una variable sin inicializar.
- **Aviso estricto:** Sobre el estilo de código se daban en los cambios de PHP 4 a PHP 5. Es difícil que se den.

# Manejo de Errores

- Directivas a nivel de php.ini

Directiva	Descripción	Valor recomendado en producción
<code>error_reporting</code>	Define qué tipos de errores se deben reportar.	<code>E_ALL &amp; ~E_DEPRECATED &amp; ~E_NOTICE</code>
<code>display_errors</code>	Muestra los errores en pantalla.	<code>Off</code>
<code>display_startup_errors</code>	Muestra errores ocurridos durante el arranque de PHP.	<code>Off</code>
<code>log_errors</code>	Guarda los errores en un archivo de log.	<code>On</code>
<code>error_log</code>	Ruta del archivo donde se guardan los errores.	<code>/var/log/php_errors.log</code>
<code>html_errors</code>	Muestra errores con formato HTML (útil en desarrollo).	<code>Off</code>
<code>ignore_repeated_errors</code>	Evita registrar errores repetidos.	<code>On</code>
<code>report_memleaks</code>	Informa sobre fugas de memoria.	<code>On</code>

# Modos de error\_reporting

- **E\_ALL**: todos los errores excepto los estrictos.
- **E\_PARSE**: Errores de análisis.
- **E\_ERROR**: Errores fatales.
- **E\_WARNING**: Advertencias.
- **E\_NOTICE**: Avisos.
- **E\_STRICT**: Avisos estrictos.

# La función die

- **die** (mensaje): Se coloca con instrucciones que puedan generar errores:
  - Por ejemplo: si vamos a abrir un fichero y no existe.
  - La forma de utilizarla sería algo así:
    - `fopen(fichero, modo) or die('se ha producido un error al abrir el fichero');`

# Gestión de Excepciones

- Esquema de las Excepciones:

```
try {  
    // Código que puede generar excepciones.  
} catch (ClaseExcepcion1 $exc1){  
    // Procesamiento de las excepciones de la clase 1  
} [catch (ClaseExcepcion2 $exc2) {  
    // Procesamiento de las excepciones de la clase 2.  
}]
```

# Gestión de las Excepciones

- Las excepciones las podemos provocar cuando se dé una situación anormal en nuestro código.
  - Con `throw new Exception($mensaje, $codigo)`
- Al lanzar la Excepción el código termina y ejecuta el código del constructor de la clase `Exception` con los parámetros que la hayamos pasado.

# La clase Exception

```
class Exception {  
    protected $message = 'Unknow exception';  
    protected $code = 0;  
    protected $file;  
    protected $line;  
  
    function __construct ($message=null, $code=0);  
    final function getMessage();  
    final function getCode();  
    final function getFile();  
    final function getLine();  
  
    function __toString(); → Redefinible.  
}
```

Podemos crear nuestras propias excepciones heredando de la clase Exception.



# Ejemplo

```
<?php
namespace App\data;

use PDO;
use PDOException;
use Exception;

class BD_Error_Recuperable extends Exception
{}

class ConexionBD {
    private string $servidorBD;
    private string $baseDatos;
    private string $usuarioBD;
    private string $pclave;
    private PDO $conexion;

    public const BD_ERROR_CONEXION = 805;

    public function __construct(
        string $servidor,
        string $bd,
        string $usuario,
        string $pclave
    ){
        $this->servidorBD = $servidor;
        $this->baseDatos = $bd;
        $this->usuarioBD = $usuario;
        $this->pclave = $pclave;
```

```
try {
    $dsn = "mysql:host={$this->servidorBD};dbname={$this->baseDatos};charset=utf8mb4";
    $this->conexion = new PDO($dsn, $this->usuarioBD, $this->pclave, [
        PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
        PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC
    ]);
} catch (PDOException $e) {
    throw new BD_Error_Recuperable(
        "Conexión fallida con el servidor {$this->servidorBD}: " . $e->getMessage(),
        self::BD_ERROR_CONEXION
    );
}

public function getConexion(): PDO {
    return $this->conexion;
}
}
```