DAWS – 2º

Autoload en PHP

JESÚS MOLINA HERNÁNDEZ
jmolina@florida-uni.es

Florida
Secundària

Índice

Introducción al Autoloading

¿Qué es el autoloading?

- **Es un mecanismo que carga automáticamente las clases de PHP cuando se instancian, sin necesidad de require o include manual.**

Beneficios del Autoloading

- Permite gestionar grandes proyectos sin preocuparse por incluir archivos manualmente.
- Mejora la legibilidad y mantenimiento del código.
- Facilita la integración con paquetes de terceros.
- Evita errores de "Clase no encontrada".

Jesús Molina Hernández – jmolina@florida-uni.es

Por qué es necesario el Autoloading

Sin autoloading, tendrías que incluir cada archivo que contiene clases manualmente.

```php
require 'Clases/Usuario.php';
require 'Clases/Producto.php';

$usuario = new Usuario();
$producto = new Producto();
```


Jesús Molina Hernández – jmolina@florida-uni.es

Autoloading tradicional con require y include

require vs include:

- *require*: Detiene el script si el archivo no se encuentra.
- *include*: Da una advertencia si el archivo no se encuentra, pero sigue ejecutando el script.

Limitaciones del enfoque tradicional

- No es eficiente para proyectos grandes.
- No se adapta bien a estructuras de directorios complejas

Jesús Molina Hernández – jmolina@florida-uni.es

Autoloading con spl_autoload_register()

spl_autoload_register():

- Es una función nativa de PHP que te permite registrar una o más funciones o métodos que se utilizarán para cargar automáticamente las clases o interfaces cuando se necesiten.

```php
spl_autoload_register(function ($class_name) {
    include 'Clases/' . $class_name . '.php';
});

$usuario = new Usuario(); // Cargará automáticamente Clases/Usuario.php
```


Jesús Molina Hernández – jmolina@florida-uni.es

Autoloading con spl_autoload_register()

¿Cómo funciona spl_autoload_register()?

- Cada vez que intentas instanciar una clase o usar una interfaz que no ha sido cargada previamente, PHP llama automáticamente a las funciones de autoloading registradas usando spl_autoload_register(). Estas funciones son responsables de cargar el archivo correcto en función del nombre de la clase o la interfaz.

```php
spl_autoload_register(function ($class_name) {
    include 'Clases/' . $class_name . '.php';
});

spl_autoload_register(function ($class_name) {
    include 'Modelos/' . $class_name . '.php';
});

$usuario = new Usuario(); // Se buscará en 'Clases/Usuario.php'
$producto = new Producto(); // Se buscará en 'Modelos/Producto.php'
```


Jesús Molina Hernández – jmolina@florida-uni.es

Autoloading con spl_autoload_register()

Funcionando con namespaces: Debes convertir el nombre del namespace en una ruta de archivo.

```php
// Autoloader que convierte el namespace en una ruta de archivo
spl_autoload_register(function ($class_name) {
    // Reemplazar los backslashes (\) por slashes (/)
    $class_name = str_replace( search: '\\', replace: '/', $class_name);

    $path = $class_name . '.php';
    if (file_exists($path)) {
        include $path;
    }
});

$usuario = new \App\Clases\Usuario(); // Cargará
```

```php
namespace App\Clases;

class Usuario {
    public function __construct() {
        echo "Clase Usuario cargada.";
    }
}
```

Este código estará en otro fichero

Jesús Molina Hernández – jmolina@florida-uni.es

Autoloading con spl_autoload_register()

Deberemos realizar funciones para cargar clases de todas las carpetas de nuestro código donde tengamos clases que se van a utilizar.

```php
spl_autoload_register(function ($class_name) {
    $directories = [
        'Clases/',
        'Modelos/',
        'Controladores/',
    ];

    foreach ($directories as $directory) {
        $file = $directory . $class_name . '.php';
        if (file_exists($file)) {
            include $file;
            return;
        }
    }
});
```

Ejemplo con una estructura habitual de MVC

Jesús Molina Hernández – jmolina@florida-uni.es

Estandarización del Autoloading (PSR-4)

¿Qué es PSR-4?

- PSR-4 es una **recomendación estándar** del **PHP-FIG (PHP Framework Interoperability Group)** para la **carga automática de clases** en PHP. Es uno de los **"PSR"** (PHP Standards Recommendations), en este caso relacionado con el autoloading.
- PSR-4 define cómo deben estar organizados los **namespaces** y las **rutas de los archivos** para que se puedan cargar automáticamente las clases sin tener que especificar cada archivo manualmente.

Jesús Molina Hernández – jmolina@florida-uni.es

Estandarización del Autoloading (PSR-4)

Reglas clave de PSR-4:

- **El namespace de la clase debe coincidir con la estructura de directorios**.
- **El nombre de la clase debe coincidir con el nombre del archivo** (incluyendo las mayúsculas/minúsculas).
- **El prefijo de namespace base** se asigna a una carpeta base.

Jesús Molina Hernández – jmolina@florida-uni.es

Estandarización del Autoloading (PSR-4)

Ejemplo de estructura de carpetas de tu proyecto

```
/proyecto
    /src
        /Clases
            Usuario.php
        /Modelos
            Producto.php
        /Controladores
            TareasController.php
    composer.json
```

- La clase **Usuario** tiene el namespace **App\Clases**, lo que corresponde a la ruta **src/Clases/Usuario.php**.
- La clase **Producto** tiene el namespace **App\Modelos**, lo que corresponde a la ruta **src/Modelos/Producto.php**.
- La clase **TareasController** tiene el namespace **App\Controladores**, lo que corresponde a la ruta **src/Controladores/TareasController.php**.

Ficheros con las definiciones de las clases

```php
// src/Clases/Usuario.php
namespace App\Clases;

class Usuario {
    // ...
}

// src/Modelos/Producto.php
namespace App\Modelos;

class Producto {
    // ...
}

// src/Controladores/TareasController.php
namespace App\Controladores;

class TareasController {
    // ...
}
```


Jesús Molina Hernández – jmolina@florida-uni.es

Estandarización del Autoloading (PSR-4)

Solución ideal: Usar el **Autoload de Composer**. Lo veremos en la presentación de Composer.

Jesús Molina Hernández – jmolina@florida-uni.es