Estructura

La función de los ficheros y directorios instalados con ZWF es la siguiente:

- `base/` Contiene las clases del framework. No es necesario modificarlo.
- `cfg/` Directorio donde colocar los ficheros de configuración.
- `controllers/` Directorio donde colocar los controladores de la aplicación.
- `lib/` Directorio de libre disposición.
- `models/` Directorio donde colocar el modelo de datos de la aplicación.
- `res/` Directorio donde colocar JS, CSS, imágenes y otros recursos públicos. Accesible desde fuera.
- `views/` Directorio donde colocar las vistas de la aplicación.
- `.htaccess` Directivas de Apache 2
- `index.php` Controlador frontal

Módulos y espacios de nombres

Las diferentes clases y ficheros de configuración de ZWF están agrupados por módulos. **Un módulo, pues, es un conjunto de clases opcionalmente unido a un fichero de configuración**.

ZWF en su instalación predeterminada tiene tres módulos:

- El módulo **core** contiene las clases básicas de ZWF.
- El módulo **dev** contiene las clases de ayuda al desarrollo.
- El módulo **data-access** contiene las clases de acceso a la base de datos.

Los módulos, a su vez, siempre se agrupan por **espacios de nombres**. El único espacio de nombres que se usa en ZWF es `zfx` y los tres módulos suministrados están bajo dicho espacio.

El directorio `base/` contiene tantos subdirectorios como espacios de nombres haya disponibles. En la instalación predeterminada solo encontraremos `base/zfx/`.

En el interior de cada directorio correspodiente a un espacio de nombres hay tantos subdirectorios como módulos haya definidos. En la instalación predeterminada de ZWF, dentro de `base/zfx/`, encontraremos los directorios correspondientes a los tres módulos disponibles:

```
base/zfx/core/
base/zfx/dev/
base/zfx/data-access/
```

Todas las clases de un módulo están definidas bajo el espacio de nombres al que pertenece dicho módulo. Cuando vayamos a invocar una clase de ZWF, en la mayoría de los casos, será necesario anteponer el espacio de nombres o usar `use`.

Es posible añadir nuevos espacios de nombres y módulos en función de las necesidades del proyecto.

Los módulos son fácimente desactivables. Un módulo desactivado es ignorado por el sistema y no se carga su configuración ni se incluye en el sistema de autocarga de clases. El módulo **core** no se puede desactivar y su carga es obligatoria.

CSS, JavaScript y otros recursos

El directorio `res/` está configurado en `.htaccess` para ser accesible públicamente. Se recomienda que el CSS, JavaScript, imágenes, vídeos, etc. de la aplicación se coloquen bajo dicho directorio.

Biblioteca de clases

A continuación se describen algunas clases suministradas con ZWF y su función. La funcionalidad se ha reducido al mínimo, siguiendo el principio de implementar sólo aquello que realmente se necesita.

La biblioteca de ZWF está documentada siguiendo el convenio PHPDocumentor y por lo tanto se puede extraer la documentación del API mediante este método.

Módulo core

- La clase `HtmlTools` contiene métodos estáticos que facilitan la creación de elementos HTML como tablas o `<select>`
- La clase `Paginator` gestiona el control y generado de código HTML para dibujar un paginador. Es altamente personalizable.
- La clase `StrFilter` contiene numersos métodos de ayuda para operar con cadenas UTF-8. **Es muy recomendable su uso.**
- La clase `StrValidator` contiene métodos de validación de tipos de datos comunes. Algunos son un stub, como el del email, que es elemental y necesitaría ser desarrollado en profundidad para cumplir con el RFC.

- La clases `Mat` y `Sys` son stubs con la intención de ser desarrolladas en un futuro. Actualmente contienen un método estático para el redondeo de euros a dos decimales y un método para obtener una ruta local respectivamente.

Módulo data-access

- La clase `DataTools` contiene herramientas útiles para el trabajo con la BD.

Biblioteca de terceros

Además de los controladores y del modelo de datos, la aplicaciones normalmente hacen uso de otras clases. Se recomienda usar el directorio `lib/` para alojar dichas clases.

Dicho directorio está bajo el sistema de carga automática de clases. Para que funcione la clase y el fichero deben tener el mismo nombre.

Revision #1
Created 4 November 2025 20:24:31 by Admin
Updated 4 November 2025 20:25:39 by Admin