

# Puesta a punto

Todo lo que te surga para completar, añadir, modificar tu instalación siguiendo las directrices de Directadmin

- [Cómo compilar una extensión o módulo de PHP en Directadmin](#)

# Cómo compilar una extensión o módulo de PHP en Directadmin

## Introducción

Una buena práctica que sigo en mis proyectos de Laravel, es la de añadir en el `composer.json` del proyecto todo aquello que necesito, inclusive las extensiones, porque no todos los paneles, no todos los SO o no todas las distribuciones, es decir, no todas las instalaciones de de un servidor o equipo de desarrollo, tiene los mismos componentes por defecto.

En el caso de hoy al migrar un despliegue de Laravel de cPanel a un DirectAdmin, cuando fui a hacer un `composer update` saltó un error indicándome la necesidad de tener la extensión `pcntl`.

Gracias a las buenas prácticas y los principios SOLID de programación, es fácil evitar un disgusto de porque no funciona, y donde está el fallo.

## Composer.json

```
{  
  ...  
  "require": {  
    "php": "^8.1",  
    "ext-pdo": "*",  
    "ext-redis": "*",  
    "ext-zlib": "*",  
    "ext-pcntl": "*",  
    ...  
  }  
}
```

# Como instalar la extensión en PHP en un servidor Directadmin

DirectAdmin tiene un número determinado de extensiones que desde la administración se pueden incluir en todas las instalaciones o desactivarlas. Pero también tiene un mecanismo sencillo para incorporar módulos de PHP a nuestra instalación, definidos en su documentación, [Add a custom module to PHP Custombuild](#)

## Localizar las configuraciones de PHP

```
da build used_configs | grep 'configure\.php'
```

PHP 8.3 configuration file: /usr/local/directadmin/custombuild/custom/php/configure.php83

PHP 8.2 configuration file: /usr/local/directadmin/custombuild/custom/php/configure.php82

PHP 5.6 configuration file: /usr/local/directadmin/custombuild/configure/php/configure.php56

PHP 7.4 configuration file: /usr/local/directadmin/custombuild/configure/php/configure.php74

## Usar el override de Directadmin

```
VERSION=82  
cd /usr/local/directadmin/custombuild  
mkdir -p custom/php  
cp -fp "configure/php/configure.php$VERSION" "custom/php/configure.php$VERSION"
```

Después deberás añadir el código necesario en el fichero `custom/php/configure.php$VERSION`, para que en la compilación de PHP se añada el módulo.

Verificamos:

```
cat configuration file: /usr/local/directadmin/custombuild/custom/php/configure.php82  
...  
[-enable-mbstring \  
[-enable-intl \  
--enable-pcntl
```

Puedes usar esto mismo para usarlos en tantas versiones como desees.



Ahora DirectAdmin ya soporta hasta 8 versiones de PHP pero recuerda, que no es buena idea añadirlo a todo por muchas razones. Carga innecesaria, versiones antiguas que igual no lo soporte, y forzar al usuario a que actualice su versión a las actuales.

# Completar el trabajo

## Compilar

```
da build php
```

## Reiniciar

```
systemctl restart httpd  
systemctl restart nginx // Si no usas httpd  
systemctl restart php-fpm$VERSION
```

## Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).