

Gestión de Versiones de PHP en Raspberry Pi

Contexto

Cuando tienes múltiples versiones de PHP instaladas en tu Raspberry Pi (especialmente usando el repositorio de [Ondrej](#)), necesitas poder cambiar entre ellas según los requisitos de cada proyecto.

Verificación inicial

Antes de configurar cualquier opción, verifica el estado actual:

```
# Ver versiones de PHP instaladas
ls /usr/bin/php*

# Ver paquetes PHP instalados
apt list --installed | grep php

# Verificar versión actual
php -v
```

Ejemplo: Instalación de PHP 8.3 (si no está presente)

“ Es un ejemplo, no es para un copy & paste

```
# Actualizar repositorios
sudo apt update
```

```
# Instalar PHP 8.3 y módulos esenciales
sudo apt install php8.3 php8.3-cli php8.3-common php8.3-curl php8.3-mbstring php8.3-mysql
php8.3-xml

# Extensiones adicionales comunes
sudo apt install php8.3-gd php8.3-zip php8.3-intl php8.3-bcmath php8.3-sqlite3
```

Opción 1: Cambio Global con update-alternatives

Descripción

Cambia la versión predeterminada de PHP para todo el sistema usando el sistema de alternativas de Debian.

Configuración

```
# Registrar las versiones disponibles
sudo update-alternatives --install /usr/bin/php php /usr/bin/php8.3 83
sudo update-alternatives --install /usr/bin/php php /usr/bin/php8.4 84

# Seleccionar la versión deseada
sudo update-alternatives --config php
```

Uso

```
# Cambiar versión interactivamente
sudo update-alternatives --config php

# Verificar cambio
php -v
```

Ventajas

- Cambio persistente entre sesiones

- Fácil de cambiar cuando necesites
- Afecta a todos los scripts y herramientas que usan `php`

Desventajas

- Cambio global afecta todo el sistema
- Puede romper scripts que dependen de una versión específica
- Requiere privilegios administrativos

Opción 2: Configuración por Sesión/Proyecto

Descripción

Configura temporalmente una versión específica de PHP solo para la sesión actual o proyecto específico.

Método A: Alias temporal

```
# Crear alias para la sesión actual
alias php='/usr/bin/php8.3'

# Verificar
php -v

# El alias se pierde al cerrar la terminal
```

Método B: Script de configuración de proyecto

Crea un archivo `setup-php.sh` en tu directorio de proyecto:

```
#!/bin/bash
# setup-php.sh

echo "Configurando entorno PHP 8.3 para el proyecto"

# Crear alias temporales
```

```
alias php='php8.3'  
alias composer='php8.3 /usr/local/bin/composer'  
  
# Verificar configuración  
echo "Versión PHP activa:"  
php8.3 -v  
  
# Opcional: exportar variables de entorno  
export PHP_VERSION=8.3  
export PHP_BINARY=/usr/bin/php8.3  
  
# Iniciar shell con configuración  
bash
```

Uso:

```
# Ejecutar el script de configuración  
source ./setup-php.sh  
  
# 0 hacer ejecutable y ejecutar  
chmod +x setup-php.sh  
./setup-php.sh
```

Método C: Configuración en .bashrc del proyecto

```
# En el directorio del proyecto, crear .bashrc local  
echo 'alias php="php8.3"' > .bashrc  
echo 'alias composer="php8.3 /usr/local/bin/composer"' >> .bashrc  
  
# Cargar configuración  
source .bashrc
```

Ventajas

- No afecta el sistema globalmente
- Configuración específica por proyecto
- Fácil de revertir

Desventajas

- Configuración temporal (se pierde al cerrar terminal)
- Requiere configuración manual para cada proyecto
- Puede ser inconsistente entre sesiones

Opción 3: Uso Directo de Versión Específica

Descripción

Usa directamente los ejecutables específicos de cada versión sin cambiar configuraciones globales.

Uso en scripts

```
# En lugar de usar 'php', usar directamente la versión
php8.3 script.php

# Para servidor de desarrollo
php8.3 -S localhost:8000

# Para composer
php8.3 /usr/local/bin/composer install
```

Configuración de shebang en scripts PHP

```
#!/usr/bin/php8.3
<?php
// Tu código aquí
```

Crear wrapper scripts

Crea un directorio para scripts wrapper:

```
# Crear directorio para wrappers
mkdir -p ~/bin
```

```
# Crear wrapper para PHP 8.3
cat > ~/bin/php83 << 'EOF'
#!/bin/bash
exec /usr/bin/php8.3 "$@"
EOF

# Hacer ejecutable
chmod +x ~/bin/php83

# Añadir al PATH (en ~/.bashrc)
echo 'export PATH="$HOME/bin:$PATH"' >> ~/.bashrc
source ~/.bashrc
```

Uso:

```
# Usar wrapper
php83 script.php
php83 -S localhost:8000
```

Ventajas

- Control total sobre qué versión usar
- No requiere configuración global
- Explícito y claro en scripts

Desventajas

- Más verboso
- Necesitas recordar usar la versión correcta
- Herramientas como Composer pueden requerir configuración adicional

Recomendaciones por Caso de Uso

Para desarrollo personal con un proyecto principal

Usar Opción 1 (update-alternatives)

- Cambio fácil y persistente
- Bueno cuando trabajas principalmente con una versión

Para múltiples proyectos con diferentes versiones

Usar Opción 2 (configuración por proyecto)

- Flexibilidad máxima
- Cada proyecto mantiene su configuración

Para scripts de producción o automatización

Usar Opción 3 (uso directo)

- Máxima claridad y control
- Sin dependencias de configuración del sistema

Verificación y Troubleshooting

```
# Verificar versión activa
php -v

# Verificar módulos instalados
php -m

# Verificar configuración de PHP
php --ini

# Ver todas las versiones disponibles
ls -la /usr/bin/php*

# Verificar composer con versión específica
php8.3 /usr/local/bin/composer --version
```

Notas Importantes

1. **Módulos y extensiones:** Cada versión de PHP tiene sus propios módulos. Asegúrate de instalar las extensiones necesarias para cada versión.
2. **Composer:** Si usas Composer, puede necesitar configuración específica:

```
# Reinstalar composer para PHP 8.3
php8.3 -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php8.3 composer-setup.php
```

3. **Servidor web:** Si usas Apache o Nginx, la configuración del servidor web es independiente de la versión de CLI.
4. **Herramientas de desarrollo:** IDEs y editores pueden necesitar configuración específica para reconocer la versión correcta.

Esta documentación te permitirá elegir la estrategia más adecuada según tus necesidades específicas de desarrollo.

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](#).

Revision #1

Created 2025-07-12 15:40:57 UTC by Abkrim

Updated 2025-07-12 15:43:01 UTC by Abkrim