

Guía Definitiva: Despliegue de Laravel 12 en Hostinger

Este Runbook detalla el proceso completo y verificado para desplegar una aplicación Laravel 12 en un entorno de hosting compartido de Hostinger, basado en lecciones aprendidas.

Fase 0: Preparación del Proyecto (Local)

1. Requisitos de Versión:

- Tu composer.json debe especificar `php >= 8.3`.
- Asegúrate de estar usando PHP 8.3 o superior en tu entorno local (XAMPP).

2. Configuración de Laravel 12:

- Asegúrate de que `routes/api.php` exista (créalo vacío si es necesario).
- Verifica que `bootstrap/app.php` cargue explícitamente las rutas de API:
- ```
->withRouting(
```
- ```
    web: __DIR__.'/../routes/web.php',
```
- ```
 api: __DIR__.'/../routes/api.php', // ¡Esencial!
```
- ```
    commands: __DIR__.'/../routes/console.php',
```
- ```
 health: '/up',
```
- ```
)
```
- Registra cualquier alias de middleware en `bootstrap/app.php` dentro de

```
->withMiddleware(...)
```

.

3. Archivo .env.example:

- Asegúrate de que contenga todas las variables de entorno necesarias para producción (BD, URL, Mail, etc.).

4. Control de Versiones (Git):

- Verifica que `.gitignore` ignore `vendor/`, `.env`, y `storage/` (excepto `storage/app/public`).
- Haz commit y push de todos tus cambios.

Fase 1: Preparación en Hostinger (hPanel)

1. Crear Base de Datos:

- Ve a Bases de datos -> Administración de Bases de Datos MySQL.
- Crea una nueva base de datos y un usuario.
- **¡CRÍTICO!** Anota el **nombre completo** de la BD y del **usuario** (con el prefijo uXXXX_) y la contraseña.

2. Establecer Versión de PHP (Web):

- Ve a Avanzado -> Configuración de PHP.
- Selecciona **PHP 8.3 o superior**.

3. Obtener Credenciales SSH:

- Ve a Avanzado -> Acceso SSH. Anota el Host, Puerto y Usuario.

Fase 2: Despliegue Inicial (SSH)

1. **Conexión SSH:** Conéctate a tu servidor.
2. **Navegación:** Ve al directorio raíz de tu dominio (ej. `cd ~/domains/tu-dominio.com/public_html`).
3. **Clonar Repositorio:** Clona tu proyecto en la carpeta actual.
4. `git clone URL_DE_TU_REPOSITORIO` .

Fase 3: Configuración del Entorno en Servidor (SSH)

1. Identificar y Configurar PHP CLI:

- **Encuentra la ruta:**
- `ls -d /opt/alt/php*/usr/bin/php`

(Elige la ruta de 8.3 o superior, ej. `/opt/alt/php83/usr/bin/php`).

- **Crea un Alias (¡MUY RECOMENDADO!):**
- `echo "alias php='/opt/alt/php83/usr/bin/php'" >> ~/.bashrc && source ~/.bashrc`
- **Verifica:** `php -v` (debe mostrar 8.3.x).

2. Instalar Dependencias: (Usa el alias php o la ruta completa si no lo creaste).

3. `php /usr/local/bin/composer install --no-dev --optimize-autoloader`

4. Configurar .env:

- Copia el ejemplo: `cp .env.example .env`
- Edita con nano .env y configura tus variables de producción (APP_ENV=production, APP_DEBUG=false, APP_URL, y las credenciales DB_ completas).
- Genera la clave: `php artisan key:generate`

5. Preparar Base de Datos (Secuencia Importante):

- **Paso A: Importar Estructura Base (.sql):**
- `mysql -u uXXXX_usuario_bd -p uXXXX_nombre_bd < tu_archivo.sql`

(O impórtalo directamente desde phpMyAdmin).

- **Paso B: Migraciones de Laravel:** (Crea tablas internas como sessions, tokens).
- `php artisan migrate --force`
- **Paso C: Seeders (Si aplica):**
- `php artisan db:seed --force`

6. Configurar Enrutamiento (Solución de "Dos Recepcionistas"):

- **Crea .htaccess en public_html (Raíz):**
- `cat > .htaccess <<'HTEOF'`
- RewriteEngine On
- RewriteCond %{REQUEST_URI} !^/public/
- RewriteRule ^(.*)\$ public/\$1 [L,QSA]
- HTEOF
- **Verifica .htaccess en public_html/public (Laravel):** Asegúrate de que exista y contenga RewriteBase /.

7. **Ajustar Permisos:**

8. `chmod -R 775 storage bootstrap/cache`

9. **Optimizar la Aplicación:**

10. `php artisan optimize`

Fase 4: Verificación y Troubleshooting

1. **Prueba la URL Raíz:** Abre `https://tu-dominio.com`. Deberías ver tu aplicación.

2. **Prueba un Endpoint de API:** Usa Postman o curl.

- `curl -I https://tu-dominio.com/api/login` (Esperado: 405 Method Not Allowed).
- `POST https://tu-dominio.com/api/login` (Esperado: 200 OK o 401 Invalid Credentials).

3. **Error 500:** Revisa `storage/logs/laravel.log`.

4. **Error 404:** Verifica ambos `.htaccess`, limpia la caché (`php artisan route:clear`), y considera desactivar la CDN de Hostinger (hPanel -> Rendimiento -> CDN) temporalmente.

5. **Error 403:** Verifica los permisos de las carpetas.

Fase 5: Despliegues de Actualización (Nuevos Módulos)

1. SSH a `public_html`.

2. `git pull origin main`

3. `php /usr/local/bin/composer install --no-dev --optimize-autoloader`

4. `php artisan migrate --force` (si hay nuevas migraciones).

5. `php artisan db:seed --force` (si hay nuevos seeders).

6. `php artisan optimize`