

Nova Página

Wiki — Instalação do Ambiente pm.ro.gov.br

> **Autor:** moraes

> **Data:** 06/05/2026

> **Objetivo:** Replicar o ambiente de produção do site WordPress `pm.ro.gov.br` em um servidor de homologação.

> **Ambiente testado:** Debian GNU/Linux 13 (trixie) — `sirius-web-hml-01`

[△](#) Diferenças entre Produção e Homologação

| Componente | Produção | Homologação (este guia) |

|-----|-----|-----|

| SO | Ubuntu 22.04.5 LTS | Debian 13 (trixie) |

| Apache | 2.4.52 | 2.4.66 [□](#) |

| PHP | 8.1.2 | 8.1.34 [□](#) (via sury.org) |

| MariaDB | 10.6.22 | 11.8.6 [△](#) (ver nota) |

| WordPress | 6.9.4 | 6.9.4 [□](#) |

| SSL/Proxy | Proxy reverso 172.16.139.3 | Sem proxy (HTTP local) |

| MariaDB bind | 0.0.0.0 (risco em prod) | 127.0.0.1 [□](#) (mais seguro) |

> **Nota MariaDB:** O Debian trixie não oferece suporte ao MariaDB 10.6 nos repositórios oficiais da MariaDB. A versão 11.8.6 é totalmente compatível com WordPress 6.9.4.

1. Pré-requisitos

- Servidor com Debian 13 (trixie) instalado

- Acesso sudo

- Conexão com a internet

2. Atualização do sistema e pacotes base

```
`` bash
```

```
sudo apt update
```

```
sudo apt install -y curl wget unzip git apt-transport-https ca-certificates gnupg lsb-release
```

```
``
```

3. Apache 2.4

Instalação

```
```bash
sudo apt install -y apache2
```
```

Ativar módulos necessários

```
```bash
sudo a2enmod rewrite headers proxy proxy_http
sudo systemctl restart apache2
```
```

Validação

```
```bash
sudo apache2ctl -v # deve exibir Apache/2.4.x
sudo systemctl is-active apache2 # deve retornar: active
```
```

4. PHP 8.1

O Debian trixie não inclui PHP 8.1 nos repositórios padrão. É necessário usar o repositório ****sury.org****:

Adicionar repositório

```
```bash
curl -sLo /tmp/sury.gpg https://packages.sury.org/php/apt.gpg
sudo install -o root -g root -m 644 /tmp/sury.gpg /etc/apt/trusted.gpg.d/sury-php.gpg
echo "deb https://packages.sury.org/php/ trixie main" | sudo tee /etc/apt/sources.list.d/sury-php.list
sudo apt update
```
```

Instalar PHP 8.1 e extensões

```
```bash
sudo apt install -y \
php8.1 \
php8.1-mysql \
php8.1-xml \
```

```
php8.1-curl \
php8.1-mbstring \
php8.1-zip \
php8.1-gd \
php8.1-intl \
php8.1-bcmath \
php8.1-soap \
php8.1-opcache \
libapache2-mod-php8.1
...
```

```
Validação
```

```
```bash  
php8.1 -v  
# PHP 8.1.x (cli) ... with Zend OPcache  
  
php8.1 -m | grep -E "^(curl|mbstring|mysqli|zip|gd|xml|intl|bcmath|soap|Zend OPcache)$"  
# deve listar todas as extensões acima  
...  
  
---
```

```
## 5. MariaDB
```

```
### Instalar (versão do repositório Debian)
```

```
```bash  
sudo apt install -y mariadb-server mariadb-client
sudo systemctl enable mariadb
sudo systemctl start mariadb
...
```

```
□ Restringir bind para 127.0.0.1 (segurança)
```

```
```bash  
sudo sed -i 's/^bind-address\s*=\s*/bind-address = 127.0.0.1/' \  
/etc/mysql/mariadb.conf.d/50-server.cnf  
  
sudo systemctl restart mariadb  
...
```

```
### Validação
```

```
```bash  
mariadb --version
sudo systemctl is-active mariadb # deve retornar: active
```

```
grep bind-address /etc/mysql/mariadb.conf.d/50-server.cnf
deve mostrar: bind-address = 127.0.0.1
...
```

---

## ## 6. Banco de dados WordPress

> ⚠️ Altere a senha `SenhaSitepm@2026` para uma senha forte antes de usar em produção.

```
```bash
```

```
sudo mysql -u root <<'SQL'
CREATE DATABASE IF NOT EXISTS sitepm
CHARACTER SET utf8mb4
COLLATE utf8mb4_unicode_ci;
```

```
CREATE USER IF NOT EXISTS 'pessoa'@'localhost'
IDENTIFIED BY 'SenhaSitepm@2026';
```

```
GRANT ALL PRIVILEGES ON sitepm.* TO 'pessoa'@'localhost';
FLUSH PRIVILEGES;
SQL
...
```

Validação

```
```bash
```

```
sudo mysql -u root -e "SHOW DATABASES LIKE 'sitepm';"
deve listar o banco sitepm
...
```

---

## ## 7. WordPress 6.9.4

### ### Download e extração

```
```bash
```

```
cd /tmp
curl -O https://wordpress.org/wordpress-6.9.4.tar.gz
tar -xzf wordpress-6.9.4.tar.gz
sudo rsync -a --delete wordpress/ /var/www/html/
sudo chown -R www-data:www-data /var/www/html
sudo find /var/www/html -type d -exec chmod 755 {} \;
sudo find /var/www/html -type f -exec chmod 644 {} \;
rm -rf /tmp/wordpress /tmp/wordpress-6.9.4.tar.gz
...
```

Criar wp-config.php

```
```bash
sudo cp /var/www/html/wp-config-sample.php /var/www/html/wp-config.php
```

```
sudo sed -i \
"s/database_name_here/sitepm; \
s/username_here/pessoa; \
s/password_here/SenhaSitepm@2026; \
s/'localhost'/'127.0.0.1'/" \
/var/www/html/wp-config.php
```
```

⚠ Inserir bloco de proxy reverso (CRÍTICO)

Este bloco corrige o loop de redirecionamento HTTPS (incidente 28/04/2026).

****Nunca remova este bloco se o ambiente usar proxy reverso.****

Insira manualmente no `wp-config.php`, ****antes**** da linha `require_once ABSPATH . 'wp-settings.php`:

```
```php
/* Detecta HTTPS quando atrás de proxy reverso */
if (isset($_SERVER['HTTP_X_FORWARDED_PROTO']) &&
'https' === strtolower($_SERVER['HTTP_X_FORWARDED_PROTO'])) {
$_SERVER['HTTPS'] = 'on';
}
```
```

Ou via linha de comando:

```
```bash
sudo sed -i "/require_once ABSPATH . 'wp-settings.php';/i \\
V* Detecta HTTPS quando atras de proxy reverso *\\
if (isset(\\$_SERVER['HTTP_X_FORWARDED_PROTO']) &&\\
'https' === strtolower(\\$_SERVER['HTTP_X_FORWARDED_PROTO'])) {\\
\\$_SERVER['HTTPS'] = 'on';\\
}" /var/www/html/wp-config.php
```
```

Gerar chaves secretas

Acesse: <https://api.wordpress.org/secret-key/1.1/salt/>

Substitua o bloco de chaves no `wp-config.php`.

Permissões finais

```
```bash
sudo chmod 640 /var/www/html/wp-config.php
sudo chown www-data:www-data /var/www/html/wp-config.php
```
```

8. VirtualHost Apache

```
```bash
sudo bash -c 'cat > /etc/apache2/sites-available/wordpress.conf <<EOF
<VirtualHost *:80>
ServerName localhost
DocumentRoot /var/www/html

<Directory /var/www/html>
Options FollowSymLinks
AllowOverride All
Require all granted
</Directory>

ErrorLog \${APACHE_LOG_DIR}/error.log
CustomLog \${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
EOF'
```

```
sudo a2dissite 000-default.conf
sudo a2ensite wordpress.conf
sudo systemctl reload apache2
```
```

Validação

```
```bash
curl -s -o /dev/null -w "%{http_code}" http://localhost/
302 = WordPress aguardando configuração inicial (esperado)
200 = WordPress já configurado
```
```

9. Segurança

fail2ban (proteção brute-force)

```
```bash
sudo apt install -y fail2ban
```

```
sudo systemctl enable fail2ban
sudo systemctl start fail2ban
sudo systemctl is-active fail2ban # deve retornar: active
...
```

### auditd (auditoria de syscalls)

```
```bash
sudo apt install -y auditd
sudo systemctl enable auditd
sudo systemctl start auditd
sudo systemctl is-active auditd # deve retornar: active
...
---
```

10. Ferramentas de diagnóstico

```
```bash
sudo apt install -y htop ncd net-tools
...

```

## 11. Finalização — WP Admin

Acesse: `http://<IP-DO-SERVIDOR>/wp-admin/install.php``

Complete o assistente de instalação do WordPress e em seguida instale os plugins abaixo via **Plugins → Adicionar novo**:

### Plugins (replicar da produção)

Plugin	Função
elementor	Page builder
elementor-pro	Page builder Pro (licença necessária)
astra-sites	Importador de templates Astra
astra-widgets	Widgets do tema Astra
all-in-one-wp-migration	Usar para importar backup da produção
wps-hide-login	Ocultar <code>/wp-login.php`</code>
wp-mail-smtp	Relay SMTP de e-mail
wp-seopress	SEO
wpfront-user-role-editor	Permissões de usuários
loginpress	Personalização da tela de login
nextgen-gallery	Galeria de imagens
hide-admin-menu	Restrição de menu administrativo

header-footer-code-manager	Código no header/footer
header-footer-elementor	Header e footer via Elementor
custom-css-js	CSS/JS customizado
duplicate-page	Duplicar páginas
duplicate-menu	Duplicar menus
classic-editor	Editor clássico (desativa Gutenberg)
re-add-underline-justify	Botões de formatação extra
ele-custom-skin	Skin para Elementor

### ### Tema

1. Instalar **Astra** (tema pai)
2. Instalar **astra-child** (tema filho)
3. Ativar o **astra-child**

### ### Arquivos customizados (copiar da produção)

```
```bash
# Copiar do servidor de produção:
scp moraes@172.16.139.59:/var/www/html/hardsec.php /var/www/html/
scp moraes@172.16.139.59:/var/www/html/index.js /var/www/html/
sudo chown www-data:www-data /var/www/html/hardsec.php /var/www/html/index.js
```
```

---

### ## 12. Importar banco da produção (via all-in-one-wp-migration)

1. No servidor de **produção**: Plugins → All-in-One WP Migration → Exportar → Arquivo
2. Baixe o arquivo `.wordpress` gerado
3. No servidor de **homologação**: Plugins → All-in-One WP Migration → Importar → Arquivo
4. Após a importação, atualizar as URLs no banco:

```
```bash
sudo mysql -u root sitepm <<'SQL'
UPDATE wp_options SET option_value = 'http://localhost'
WHERE option_name IN ('siteurl', 'home');
SQL
```
```

---

### ## 13. Validação final do ambiente

```
```bash
echo "=== Serviços ==="
sudo systemctl is-active apache2 mariadb fail2ban auditd
```

```
echo "=== Versões ==="  
sudo apache2ctl -v | head -1  
php8.1 -v | head -1  
mariadb --version
```

```
echo "=== Banco ==="  
sudo mysql -u root -e "SHOW DATABASES LIKE 'sitepm';"
```

```
echo "=== WordPress ==="  
ls /var/www/html/wp-login.php
```

```
echo "=== HTTP ==="  
curl -s -o /dev/null -w "Status: %{http_code}\n" http://localhost/  
`
```

Revision #1

Created 6 May 2026 17:01:46 by De Moraes

Updated 6 May 2026 17:02:13 by De Moraes