

# Instalação e Configurações WordPress Docker

- [Instalação WordPress Docker \(Bitnami\)](#)
- [Como Limpar Cache WordPress: 6 Métodos Explicados](#)
- [Compressão GZIP para melhor desempenho do Wordpress](#)
- [Como Otimizar Seu Site WordPress Em 11 Passos - Guia 2024](#)

# Instalação WordPress Docker (Bitnami)

Link: <https://github.com/bitnami/containers/tree/main/bitnami/wordpress#how-to-use-this-image>

## Bitnami package for WordPress

## What is WordPress?

“WordPress is the world's most popular blogging and content management platform. Powerful yet simple, everyone from students to global corporations use it to build beautiful, functional websites.

### Overview of WordPress

## TL;DR

```
docker run --name wordpress bitnami/wordpress:latest
```

**Warning:** This quick setup is only intended for development environments. You are encouraged to change the insecure default credentials and check out the available configuration options in the Environment Variables section for a more secure deployment.

## Why use Bitnami Images?

- Bitnami closely tracks upstream source changes and promptly publishes new versions of this image using our automated systems.
- With Bitnami images the latest bug fixes and features are available as soon as possible.

- Bitnami containers, virtual machines and cloud images use the same components and configuration approach - making it easy to switch between formats based on your project needs.
- All our images are based on **minideb** -a minimalist Debian based container image that gives you a small base container image and the familiarity of a leading Linux distribution- or **scratch** -an explicitly empty image-.
- All Bitnami images available in Docker Hub are signed with Notation. Check this post to know how to verify the integrity of the images.
- Bitnami container images are released on a regular basis with the latest distribution packages available.

Looking to use WordPress in production? Try VMware Tanzu Application Catalog, the commercial edition of the Bitnami catalog.

# How to deploy WordPress in Kubernetes?

Deploying Bitnami applications as Helm Charts is the easiest way to get started with our applications on Kubernetes. Read more about the installation in the Bitnami WordPress Chart GitHub repository.

Bitnami containers can be used with Kubeapps for deployment and management of Helm Charts in clusters.

## Why use a non-root container?

Non-root container images add an extra layer of security and are generally recommended for production environments. However, because they run as a non-root user, privileged tasks are typically off-limits. Learn more about non-root containers in our docs.

## Supported tags and respective Dockerfile links

Learn more about the Bitnami tagging policy and the difference between rolling tags and immutable tags [in our documentation page](#).

You can see the equivalence between the different tags by taking a look at the `tags-info.yaml` file present in the branch folder, i.e `bitnami/ASSET/BRANCH/DISTRO/tags-info.yaml`.

Subscribe to project updates by watching the [bitnami/containers GitHub repo](#).

## Get this image

The recommended way to get the Bitnami WordPress Docker Image is to pull the prebuilt image from the [Docker Hub Registry](#).

```
docker pull bitnami/wordpress:latest
```

To use a specific version, you can pull a versioned tag. You can view the [list of available versions](#) in the Docker Hub Registry.

```
docker pull bitnami/wordpress:[TAG]
```

If you wish, you can also build the image yourself by cloning the repository, changing to the directory containing the Dockerfile and executing the `docker build` command. Remember to replace the `APP`, `VERSION` and `OPERATING-SYSTEM` path placeholders in the example command below with the correct values.

```
git clone https://github.com/bitnami/containers.git
cd bitnami/APP/VERSION/OPERATING-SYSTEM
docker build -t bitnami/APP:latest .
```

## How to use this image

WordPress requires access to a MySQL or MariaDB database to store information. We'll use the [Bitnami Docker Image for MariaDB](#) for the database requirements.

## Using the Docker Command Line

## Step 1: Create a network

```
docker network create wordpress-network
```

## Step 2: Create a volume for MariaDB persistence and create a MariaDB container

```
$ docker volume create --name mariadb_data
docker run -d --name mariadb \
  --env ALLOW_EMPTY_PASSWORD=yes \
  --env MARIADB_USER=bn_wordpress \
  --env MARIADB_PASSWORD=bitnami \
  --env MARIADB_DATABASE=bitnami_wordpress \
  --network wordpress-network \
  --volume mariadb_data:/bitnami/mariadb \
  bitnami/mariadb:latest
```

## Step 3: Create volumes for WordPress persistence and launch the container

```
$ docker volume create --name wordpress_data
docker run -d --name wordpress \
  -p 8080:8080 -p 8443:8443 \
  --env ALLOW_EMPTY_PASSWORD=yes \
  --env WORDPRESS_DATABASE_USER=bn_wordpress \
  --env WORDPRESS_DATABASE_PASSWORD=bitnami \
  --env WORDPRESS_DATABASE_NAME=bitnami_wordpress \
  --network wordpress-network \
  --volume wordpress_data:/bitnami/wordpress \
  bitnami/wordpress:latest
```

Access your application at `http://your-ip/`

# Run the application using Docker Compose

```
curl -sSL https://raw.githubusercontent.com/bitnami/containers/main/bitnami/wordpress/docker-compose.yml > dc
docker-compose up -d
```

Please be aware this file has not undergone internal testing. Consequently, we advise its use exclusively for development or testing purposes. For production-ready deployments, we highly

recommend utilizing its associated [Bitnami Helm chart](#).

If you detect any issue in the `docker-compose.yml` file, feel free to report it or contribute with a fix by following our [Contributing Guidelines](#).

# Persisting your application

If you remove the container all your data will be lost, and the next time you run the image the database will be reinitialized. To avoid this loss of data, you should mount a volume that will persist even after the container is removed.

For persistence you should mount a directory at the `/bitnami/wordpress` path. If the mounted directory is empty, it will be initialized on the first run. Additionally you should mount a volume for persistence of the MariaDB data.

The above examples define the Docker volumes named `mariadb_data` and `wordpress_data`. The WordPress application state will persist as long as volumes are not removed.

To avoid inadvertent removal of volumes, you can mount host directories as data volumes. Alternatively you can make use of volume plugins to host the volume data.

## Mount host directories as data volumes with Docker Compose

This requires a minor change to the `docker-compose.yml` file present in this repository:

```
mariadb:
  ...
  volumes:
-   - 'mariadb_data:/bitnami/mariadb'
+   - '/path/to/mariadb-persistence:/bitnami/mariadb'
  ...
wordpress:
  ...
  volumes:
-   - 'wordpress_data:/bitnami/wordpress'
+   - '/path/to/wordpress-persistence:/bitnami/wordpress'
  ...
-volumes:
- mariadb_data:
-   driver: local
```

- wordpress\_data:
- driver: local

“ NOTE: As this is a non-root container, the mounted files and directories must have the proper permissions for the UID `1001` .

# Mount host directories as data volumes using the Docker command line

## Step 1: Create a network (if it does not exist)

```
docker network create wordpress-network
```

## Step 2. Create a MariaDB container with host volume

```
docker run -d --name mariadb \  
--env ALLOW_EMPTY_PASSWORD=yes \  
--env MARIADB_USER=bn_wordpress \  
--env MARIADB_PASSWORD=bitnami \  
--env MARIADB_DATABASE=bitnami_wordpress \  
--network wordpress-network \  
--volume /path/to/mariadb-persistence:/bitnami/mariadb \  
bitnami/mariadb:latest
```

“ NOTE: As this is a non-root container, the mounted files and directories must have the proper permissions for the UID `1001` .

## Step 3. Create the WordPress container with host volumes

```
docker run -d --name wordpress \  
-p 8080:8080 -p 8443:8443 \  
--env ALLOW_EMPTY_PASSWORD=yes \  
--env WORDPRESS_DATABASE_USER=bn_wordpress \  
--env WORDPRESS_DATABASE_PASSWORD=bitnami \  
--env WORDPRESS_DATABASE_NAME=bitnami_wordpress \  
--network wordpress-network \  
--volume /path/to/wordpress-persistence:/bitnami/wordpress \  
bitnami/wordpress:latest
```

NOTE: As this is a non-root container, the mounted files and directories must have the proper permissions for the UID `1001`.

# Configuration

## Environment variables

### Customizable environment variables

Name	Description	Default Value
<code>WORDPRESS_DATA_T O_PERSIST</code>	Files to persist relative to the WordPress installation directory. To provide multiple values, separate them with a whitespace.	<code>wp-config.php wp- content</code>
<code>WORDPRESS_ENABL E_HTTPS</code>	Whether to enable HTTPS for WordPress by default.	<code>no</code>
<code>WORDPRESS_BLOG_ NAME</code>	WordPress blog name.	<code>"User's blog"</code>
<code>WORDPRESS_SCHEM E</code>	Scheme to generate application URLs. Deprecated by <code>WORDPRESS_ENABLE_HTTPS</code> .	<code>http</code>
<code>WORDPRESS_HTACC ESS_OVERRIDE_NON E</code>	Set the Apache <code>AllowOverride</code> variable to <code>None</code> . All the default directives will be loaded from <code>/opt/bitnami/wordpress/wordpress-htaccess.conf</code> .	<code>yes</code>
<code>WORDPRESS_ENABL E_HTACCESS_PERSIS TENCE</code>	Persist the custom changes of the htaccess. It depends on the value of <code>WORDPRESS_HTACCESS_OVERRIDE_NONE</code> , when <code>yes</code> it will persist <code>/opt/bitnami/wordpress/wordpress-htaccess.conf</code> if <code>no</code> it will persist <code>/opt/bitnami/wordpress/.htaccess</code> .	<code>no</code>
<code>WORDPRESS_RESET_ DATA_PERMISSIONS</code>	Force resetting ownership/permissions on persisted data when initializing, otherwise it assumes the ownership/permissions are correct. Ignored when running as non-root.	<code>no</code>
<code>WORDPRESS_TABLE_ PREFIX</code>	Table prefix to use in WordPress.	<code>wp_</code>
<code>WORDPRESS_PLUGIN S</code>	List of WordPress plugins to install and activate, separated via commas. Can also be set to <code>all</code> to activate all currently installed plugins, or <code>none</code> to skip.	<code>none</code>
<code>WORDPRESS_EXTRA_ INSTALL_ARGS</code>	Extra flags to append to the WordPress 'wp core install' command call.	<code>nil</code>
<code>WORDPRESS_EXTRA_ CLI_ARGS</code>	Extra flags to append to all WP-CLI command calls.	<code>nil</code>



Name	Description	Default Value
WORDPRESS_EXTRA_WP_CONFIG_CONTENT	Extra configuration to append to wp-config.php during install.	nil
WORDPRESS_SKIP_BOOTSTRAP	Whether to perform initial bootstrapping for the application.	no
WORDPRESS_AUTO_UPDATE_LEVEL	Level of auto-updates to allow for the WordPress core installation. Valid values: <code>major</code> , <code>minor</code> , <code>none</code> .	none
WORDPRESS_AUTH_KEY	Value of the AUTH_KEY	nil
WORDPRESS_SECURE_AUTH_KEY	Value of the SECURE_AUTH_KEY	nil
WORDPRESS_LOGGED_IN_KEY	Value of the LOGGED_IN_KEY	nil
WORDPRESS_NONCE_KEY	Value of the NONCE_KEY	nil
WORDPRESS_AUTH_SALT	Value of the AUTH_SALT	nil
WORDPRESS_SECURE_AUTH_SALT	Value of the SECURE_AUTH_SALT	nil
WORDPRESS_LOGGED_IN_SALT	Value of the LOGGED_IN_SALT	nil
WORDPRESS_NONCE_SALT	Value of the NONCE_SALT	nil
WORDPRESS_ENABLE_REVERSE_PROXY	Enable WordPress support for reverse proxy headers	no
WORDPRESS_ENABLE_XML_RPC	Enable the WordPress XML-RPC endpoint	no
WORDPRESS_USERNAME	WordPress user name.	user
WORDPRESS_PASSWORD	WordPress user password.	bitnami
WORDPRESS_EMAIL	WordPress user e-mail address.	user@example.com
WORDPRESS_FIRST_NAME	WordPress user first name.	UserName
WORDPRESS_LAST_NAME	WordPress user last name.	LastName
WORDPRESS_ENABLE_MULTISITE	Enable WordPress Multisite configuration.	no
WORDPRESS_MULTISITE_NETWORK_TYPE	WordPress Multisite network type to enable. Valid values: <code>subfolder</code> , <code>subdirectory</code> , <code>subdomain</code> .	subdomain
WORDPRESS_MULTISITE_EXTERNAL_HTTP_PORT_NUMBER	External HTTP port for WordPress Multisite.	80

Name	Description	Default Value
WORDPRESS_MULTISITE_EXTERNAL_HTTPS_PORT_NUMBER	External HTTPS port for WordPress Multisite.	443
WORDPRESS_MULTISITE_HOST	WordPress hostname/address. Only used for Multisite installations.	nil
WORDPRESS_MULTISITE_ENABLE_NIP_IO_REDIRECTION	Whether to enable IP address redirection to nip.io wildcard DNS when enabling WordPress Multisite. This is only supported when running on an IP address with subdomain network type.	no
WORDPRESS_MULTISITE_FILEUPLOAD_MAX_K	Maximum upload file size allowed for WordPress Multisite uploads, in kilobytes.	81920
WORDPRESS_SMTP_HOST	WordPress SMTP server host.	nil
WORDPRESS_SMTP_PORT_NUMBER	WordPress SMTP server port number.	nil
WORDPRESS_SMTP_USER	WordPress SMTP server user.	nil
WORDPRESS_SMTP_FROM_EMAIL	WordPress SMTP from email.	\${WORDPRESS_SMTP_USER}
WORDPRESS_SMTP_FROM_NAME	WordPress SMTP from name.	\${WORDPRESS_FIRST_NAME} \${WORDPRESS_LAST_NAME}
WORDPRESS_SMTP_PASSWORD	WordPress SMTP server user password.	nil
WORDPRESS_SMTP_PROTOCOL	WordPress SMTP server protocol to use.	nil
WORDPRESS_DATABASE_HOST	Database server host.	\$WORDPRESS_DEFAULT_DATABASE_HOST
WORDPRESS_DATABASE_PORT_NUMBER	Database server port.	3306
WORDPRESS_DATABASE_NAME	Database name.	bitnami_wordpress
WORDPRESS_DATABASE_USER	Database user name.	bn_wordpress
WORDPRESS_DATABASE_PASSWORD	Database user password.	nil
WORDPRESS_ENABLE_DATABASE_SSL	Whether to enable SSL for database connections.	no
WORDPRESS_VERIFY_DATABASE_SSL	Whether to verify the database SSL certificate when SSL is enabled for database connections.	yes
WORDPRESS_DATABASE_SSL_CERT_FILE	Path to the database client certificate file.	nil
WORDPRESS_DATABASE_SSL_KEY_FILE	Path to the database client certificate key file.	nil

Name	Description	Default Value
<code>WORDPRESS_DATABASE_SSL_CA_FILE</code>	Path to the database server CA bundle file.	<code>nil</code>
<code>WORDPRESS_OVERRIDE_DATABASE_SETTINGS</code>	Override the database settings in persistence.	<code>no</code>

## Read-only environment variables

Name	Description	Value
<code>WORDPRESS_BASE_DIR</code>	WordPress installation directory.	<code>\${BITNAMI_ROOT_DIR}/wordpress</code>
<code>WORDPRESS_CONF_FILE</code>	Configuration file for WordPress.	<code>\${WORDPRESS_BASE_DIR}/wp-config.php</code>
<code>WP_CLI_BASE_DIR</code>	WP-CLI installation directory.	<code>\${BITNAMI_ROOT_DIR}/wp-cli</code>
<code>WP_CLI_BIN_DIR</code>	WP-CLI directory for binary files.	<code>\${WP_CLI_BASE_DIR}/bin</code>
<code>WP_CLI_CONF_DIR</code>	WP-CLI directory for configuration files.	<code>\${WP_CLI_BASE_DIR}/conf</code>
<code>WP_CLI_CONF_FILE</code>	Configuration file for WP-CLI.	<code>\${WP_CLI_CONF_DIR}/wp-cli.yml</code>
<code>WORDPRESS_VOLUME_DIR</code>	WordPress directory for mounted configuration files.	<code>\${BITNAMI_VOLUME_DIR}/wordpress</code>
<code>WORDPRESS_DEFAULT_DATABASE_HOST</code>	Default database server host.	<code>mariadb</code>
<code>PHP_DEFAULT_MEMORY_LIMIT</code>	Default PHP memory limit.	<code>512M</code>
<code>PHP_DEFAULT_POST_MAX_SIZE</code>	Default PHP post_max_size.	<code>80M</code>
<code>PHP_DEFAULT_UPLOAD_MAX_FILESIZE</code>	Default PHP upload_max_size.	<code>80M</code>
<code>WP_CLI_DAEMON_USER</code>	WP-CLI system user.	<code>daemon</code>
<code>WP_CLI_DAEMON_GROUP</code>	WP-CLI system group.	<code>daemon</code>

When you start the WordPress image, you can adjust the configuration of the instance by passing one or more environment variables either on the docker-compose file or on the `docker run` command line. Please note that some variables are only considered when the container is started for the first time. If you want to add a new environment variable:

- For docker-compose add the variable name and value under the application section in the `docker-compose.yml` file present in this repository:

```
wordpress:
  ...
  environment:
    ...
```

- For manual execution add a `--env` option with each variable and value:

```
$ docker run -d --name wordpress -p 80:8080 -p 443:8443 \
--env WORDPRESS_PASSWD=wordpress --network wordpress-network \
--volume /path/to/wordpress-pd:/var/www/html
bitnami/wordpress:latest
```

# Examples

## SMTP configuration using a Gmail account

This would be an example of SMTP configuration using a Gmail account:

- Modify the `docker-compose.yml` file present in this repository:

```
wordpress:
  ...
  environment:
    ...
```

- For manual execution:

```
$ docker run -d --name wordpress -p 80:8080 -p 443:8443 \
--env WORDPRESS_DATABASE_USER=wordpress --env WORDPRESS_DATABASE_NAME=bitnami_wordpress \
--env WORDPRESS_SMTP_HOST=smtp.gmail.com --env WORDPRESS_SMTP_PORT=587 \
--env WORDPRESS_SMTP_USER=your_email@gmail.com --env WORDPRESS_SMTP_PASSWORD=your_password \
--network wordpress-network --volume /path/to/wordpress-pd:/var/www/html
bitnami/wordpress:latest
```

## Connect WordPress container to an existing database

The Bitnami WordPress container supports connecting the WordPress application to an external database. This would be an example of using an external database for WordPress.

- Modify the `docker-compose.yml` file present in this repository:

```
wordpress:
  ...
  environment:
  -
  +

  -
  +
  ...
```

- For manual execution:

```
$ docker run -d --name wordpress \
  -p 8080:8080 -p
  --network
  --env WORDPRESS_DATABASE_HOST=wordpress_db
  --env WORDPRESS_DATABASE_PORT=3306
  --env WORDPRESS_DATABASE_NAME=wordpress
  --env WORDPRESS_DATABASE_USER=root
  --env WORDPRESS_DATABASE_PASSWORD=wordpress
  --volume wordpress_data:/var/www/html
bitnami/wordpress:latest
```

In case the database already contains data from a previous WordPress installation, you need to set the variable `WORDPRESS_SKIP_BOOTSTRAP` to `yes`. Otherwise, the container would execute the installation wizard and could modify the existing data in the database. Note that, when setting `WORDPRESS_SKIP_BOOTSTRAP` to `yes`, values for environment variables such as `WORDPRESS_USERNAME`, `WORDPRESS_PASSWORD` or `WORDPRESS_EMAIL` will be ignored. Make sure that, in this imported database, the table prefix matches the one set in `WORDPRESS_TABLE_PREFIX`.

## WP-CLI tool

The Bitnami WordPress container includes the command line interface **wp-cli** that can help you to manage and interact with your WP sites. To run this tool, please note you need use the proper system user, **daemon**.

This would be an example of using **wp-cli** to display the help menu:

- Using `docker-compose` command:

```
docker-compose exec wordpress wp help
```

- Using `docker` command:

```
docker exec wordpress wp help
```

Find more information about parameters available in the tool in the [official documentation](#).

# Logging

The Bitnami WordPress Docker image sends the container logs to `stdout`. To view the logs:

```
docker logs wordpress
```

Or using Docker Compose:

```
docker-compose logs wordpress
```

You can configure the containers [logging driver](#) using the `--log-driver` option if you wish to consume the container logs differently. In the default configuration docker uses the `json-file` driver.

# Maintenance

## Backing up your container

To backup your data, configuration and logs, follow these simple steps:

### Step 1: Stop the currently running container

```
docker stop wordpress
```

Or using Docker Compose:

```
docker-compose stop wordpress
```

### Step 2: Run the backup command

We need to mount two volumes in a container we will use to create the backup: a directory on your host to store the backup in, and the volumes from the container we just stopped so we can access the data.

```
docker run --rm -v /path/to/wordpress-backups:/backups --volumes-from wordpress busybox \
cp -a /bitnami/wordpress /backups/latest
```

# Restoring a backup

Restoring a backup is as simple as mounting the backup as volumes in the containers.

For the MariaDB database container:

```
$ docker run -d --name mariadb \
...
- --volume /path/to/mariadb-persistence:/bitnami/mariadb \
+ --volume /path/to/mariadb-backups/latest:/bitnami/mariadb \
bitnami/mariadb:latest
```

For the WordPress container:

```
$ docker run -d --name wordpress \
...
- --volume /path/to/wordpress-persistence:/bitnami/wordpress \
+ --volume /path/to/wordpress-backups/latest:/bitnami/wordpress \
bitnami/wordpress:latest
```

# Upgrade this image

Bitnami provides up-to-date versions of MariaDB and WordPress, including security patches, soon after they are made upstream. We recommend that you follow these steps to upgrade your container. We will cover here the upgrade of the WordPress container. For the MariaDB upgrade see <https://github.com/bitnami/containers/tree/main/bitnami/mariadb#upgrade-this-image>

The `bitnami/wordpress:latest` tag always points to the most recent release. To get the most recent release you can simple repull the `latest` tag from the Docker Hub with `docker pull bitnami/wordpress:latest`. However it is recommended to use tagged versions.

## Step 1: Get the updated image

```
docker pull bitnami/wordpress:latest
```

## Step 2: Stop the running container

Stop the currently running container using the command

```
docker-compose stop wordpress
```

## Step 3: Take a snapshot of the application state

Follow the steps in [Backing up your container](#) to take a snapshot of the current application state.

## Step 4: Remove the currently running container

Remove the currently running container by executing the following command:

```
docker-compose rm -v wordpress
```

## Step 5: Run the new image

Update the image tag in `docker-compose.yml` and re-create your container with the new image:

```
docker-compose up -d
```

# Customize this image

The Bitnami WordPress Docker image is designed to be extended so it can be used as the base image for your custom web applications.

## Extend this image

Before extending this image, please note there are certain configuration settings you can modify using the original image:

- Settings that can be adapted using environment variables. For instance, you can change the ports used by Apache for HTTP and HTTPS, by setting the environment variables `APACHE_HTTP_PORT_NUMBER` and `APACHE_HTTPS_PORT_NUMBER` respectively.
- [Adding custom virtual hosts](#).
- [Replacing the 'httpd.conf' file](#).
- [Using custom SSL certificates](#).



If your desired customizations cannot be covered using the methods mentioned above, extend the image. To do so, create your own image using a Dockerfile with the format below:

```
FROM bitnami/wordpress
## Put your customizations below
...
```

Here is an example of extending the image with the following modifications:

- Install the `vim` editor
- Modify the Apache configuration file
- Modify the ports used by Apache

```
FROM bitnami/wordpress

## Change user to perform privileged actions
USER 0
## Install 'vim'
RUN install_packages vim
## Revert to the original non-root user
USER 1001

## Enable mod_ratelimit module
RUN sed -i -r 's/#LoadModule ratelimit_module/LoadModule ratelimit_module/' /opt/bitnami/apache/conf/httpd.conf

## Modify the ports used by Apache by default
# It is also possible to change these environment variables at runtime
ENV APACHE_HTTP_PORT_NUMBER=8181
ENV APACHE_HTTPS_PORT_NUMBER=8143
EXPOSE 8181 8143
```

Based on the extended image, you can update the `docker-compose.yml` file present in this repository to add other features:

```
wordpress:
- image: bitnami/wordpress:latest
+ build: .
  ports:
-   - '80:8080'
-   - '443:8443'
+   - '80:8181'
+   - '443:8143'
  environment:
+   - PHP_MEMORY_LIMIT=512m
...
```

# Notable Changes

## 6.4.1-debian-11-r5

- The XML-RPC endpoint has been disabled by default. Users can manually activate via the new `WORDPRESS_ENABLE_XML_RPC` environment variable.

## 5.7.1-debian-10-r21

- The size of the container image has been decreased.
- The configuration logic is now based on Bash scripts in the *rootfs/* folder.
- Multisite support was added via `WORDPRESS_ENABLE_MULTISITE` and related environment variables.
- Plugins can be installed and activated on the first deployment via `WORDPRESS_PLUGINS`.
- Added support for limiting auto-updates to WordPress core via `WORDPRESS_AUTO_UPDATE_LEVEL`. In addition, auto-updates have been disabled by default. To update WordPress core, we recommend to swap the container image version for your deployment instead of using the built-in update functionality.
- This image now supports connecting to MySQL and MariaDB databases securely via SSL.

## 5.3.2-debian-10-r30

- The WordPress container has been migrated to a "non-root" user approach. Previously the container ran as the `root` user and the Apache daemon was started as the `daemon` user. From now on, both the container and the Apache daemon run as user `1001`. You can revert this behavior by changing `USER 1001` to `USER root` in the Dockerfile.
- Consequences:
  - The HTTP/HTTPS ports exposed by the container are now `8080/8443` instead of `80/443`.
  - Backwards compatibility is not guaranteed when data is persisted using docker or docker-compose. We highly recommend migrating the WP site by exporting its content, and importing it on a new WordPress container. In the links below you'll find some alternatives:
    - [Migrate WordPress using All-in-One WP Migration plugin](#)
    - [Migrate WordPress using VaultPress](#)
  - No writing permissions will be granted on `wp-config.php` by default.

## 5.2.1-debian-9-r9 and 5.2.1-ol-7-r9

- This image has been adapted so it's easier to customize. See the [Customize this image](#) section for more information.
- The Apache configuration volume (`/bitnami/apache`) has been deprecated, and support for this feature will be dropped in the near future. Until then, the container will enable the Apache configuration from that volume if it exists. By default, and if the configuration volume does not exist, the configuration files will be regenerated each time the container is created. Users wanting to apply custom Apache configuration files are advised to mount a volume for the configuration at `/opt/bitnami/apache/conf`, or mount specific configuration files individually.
- The PHP configuration volume (`/bitnami/php`) has been deprecated, and support for this feature will be dropped in the near future. Until then, the container will enable the PHP configuration from that volume if it exists. By default, and if the configuration volume does not exist, the configuration files will be regenerated each time the container is created. Users wanting to apply custom PHP configuration files are advised to mount a volume for the configuration at `/opt/bitnami/php/conf`, or mount specific configuration files individually.
- Enabling custom Apache certificates by placing them at `/opt/bitnami/apache/certs` has been deprecated, and support for this functionality will be dropped in the near future. Users wanting to enable custom certificates are advised to mount their certificate files on top of the preconfigured ones at `/certs`.

## 5.1.1-r28, 5.1.1-rhel-7-r31 and 5.1.1-ol-7-r30

- Users reported that they wanted to import their WordPress database from other installations. Now, in order to cover this use case, the variable `WORDPRESS_SKIP_INSTALL` can be set to avoid the container launch the WordPress installation wizard.

## 5.0.3-r20

- For performance and security reasons, Apache will set the `AllowOverride` directive to `None` by default. This means that, instead of using `.htaccess` files, all the default directives will be moved to the `/opt/bitnami/wordpress/wordpress-htaccess.conf` file. The only downside of this is the compatibility with certain plugins, which would require changes in that file (you would need to mount a modified version of `wordpress-htaccess.conf` compatible with these plugins). If you want to have the default `.htaccess` behavior, set the `WORDPRESS_HTACCESS_OVERRIDE_NONE` env var to `no`.

# 5.0.0-r0

- **wp-cli** tool is included in the Docker image. Find it at **/opt/bitnami/wp-cli/bin/wp**.

## Contributing

We'd love for you to contribute to this container. You can request new features by creating an [issue](#) or submitting a [pull request](#) with your contribution.

## Issues

If you encountered a problem running this container, you can file an [issue](#). For us to provide better support, be sure to fill the issue template.

## License

Copyright © 2024 Broadcom. The term "Broadcom" refers to Broadcom Inc. and/or its subsidiaries.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# Como Limpar Cache WordPress: 6 Métodos Explicados

Link: <https://www.hostinger.com.br/tutoriais/como-limpar-cache-wordpress>

Agosto/2024

Como Limpar Cache WordPress: 6 Métodos Explicados

Se você tem um, garantir que ele tem tempos de carregamento otimizados é essencial para a experiência do usuário e para a otimização para os mecanismos de busca (SEO). Sites com boa velocidade tendem a ter taxas de rejeição mais baixas, o que resulta em maior engajamento e conversões. O Google também favorece sites rápidos e os coloca em uma posição de destaque nas suas páginas de resultados.

A questão é que muitos componentes afetam a velocidade de um site, desde a qualidade do seu código até o seu provedor de hospedagem. Por conta disso, existem muitas maneiras de **acelerar o desempenho de um site** — e cuidar do cache costuma ser um dos caminhos mais eficazes.

Cache é o nome que se dá para o armazenamento de cópias dos dados do seu site para solicitações futuras. Ou seja, se uma pessoa acessa o seu site repetidas vezes, em vez de baixar os arquivos do servidor novamente a cada visita, o cache deixa esses arquivos salvos no navegador do visitante. Desta forma, seu conteúdo é exibido mais rapidamente.

No entanto, arquivos em cache podem expirar, o que impede os visitantes de ver a versão atualizada do seu site.

Para resolver este problema, vamos mostrar como limpar o cache no **WordPress** usando alguns métodos populares. Nós também ensinaremos como remover o cache diretamente do seu navegador, para resultados mais eficazes. Vamos começar com uma explicação do que é cache e como ele funciona.

Conteúdo

- [Limpar Cache WordPress Tutorial em Vídeo](#)
- [O Que é Cache do WordPress](#)
- [Como Limpar o Cache no WordPress](#)
  - [Limpando Cache com o LiteSpeed](#)
  - [Limpando Cache WordPress no hPanel](#)
  - [Limpando Cache Com o WP Super Cache](#)
  - [Limpando o Cache Com W3 Total Cache](#)
  - [Limpando Cache Com o WP Fastest Cache](#)
  - [Limpando Cache com a Cloudflare](#)
- [Limpando Cache no Navegador](#)
- [Como Limpar o Cache WordPress Perguntas Frequentes](#)
  - [Com Que Frequência Devo Limpar o Cache no WordPress?](#)
  - [Como Limpar o Cache Sem um Plugin do WordPress?](#)
  - [Quais São os Benefícios de Limpar o Cache?](#)

# Limpar Cache WordPress: Tutorial em Vídeo

Aprenda como limpar o cache do WordPress com nosso tutorial em vídeo para otimizar a velocidade do seu site.

<https://www.youtube.com/embed/1CWYJkdMijY?feature=oembed>

## O Que é Cache do WordPress

Um cache é um local de armazenamento temporário para conteúdo estático, como arquivos HTML, Javascript, CSS e de mídia.

Sempre que você visita um site, seu navegador envia solicitações para o servidor do site. O servidor pode precisar de alguns segundos para buscar e exibir o conteúdo solicitado, mas o cache torna esse processo bem mais rápido.

Em outras palavras, o armazenamento em cache faz o download do conteúdo estático do site e o armazena como arquivos de cache. Como resultado, seu navegador pode criar menos solicitações ao servidor, pois já possui o conteúdo armazenado em cache.

Existem dois principais sistemas de cache:

- **Armazenamento em cache do lado do cliente** – também conhecido como cache do navegador, este tipo de armazenamento em cache salva o conteúdo estático no armazenamento local do visitante. Quando um usuário faz uma visita repetida ao seu site, ele carregará mais rápido, pois a versão em cache já está armazenada no navegador.
- **Cache do lado do servidor** – tem um conceito semelhante ao cache do lado do cliente. No entanto, ele usa um servidor para armazenar os dados em cache de um site. O cache do servidor pode armazenar vários dados do site em múltiplos bancos de dados, tornando-o uma ótima solução para sites com alto tráfego.

No geral, o cache reduz as solicitações que seu servidor recebe armazenando uma versão estática do seu site — aumentando, assim, o desempenho do seu site.

Isso também reduz seu consumo da sua rede, pois processar menos solicitações requer menos largura de banda, o que pode beneficiar sites com recursos limitados.

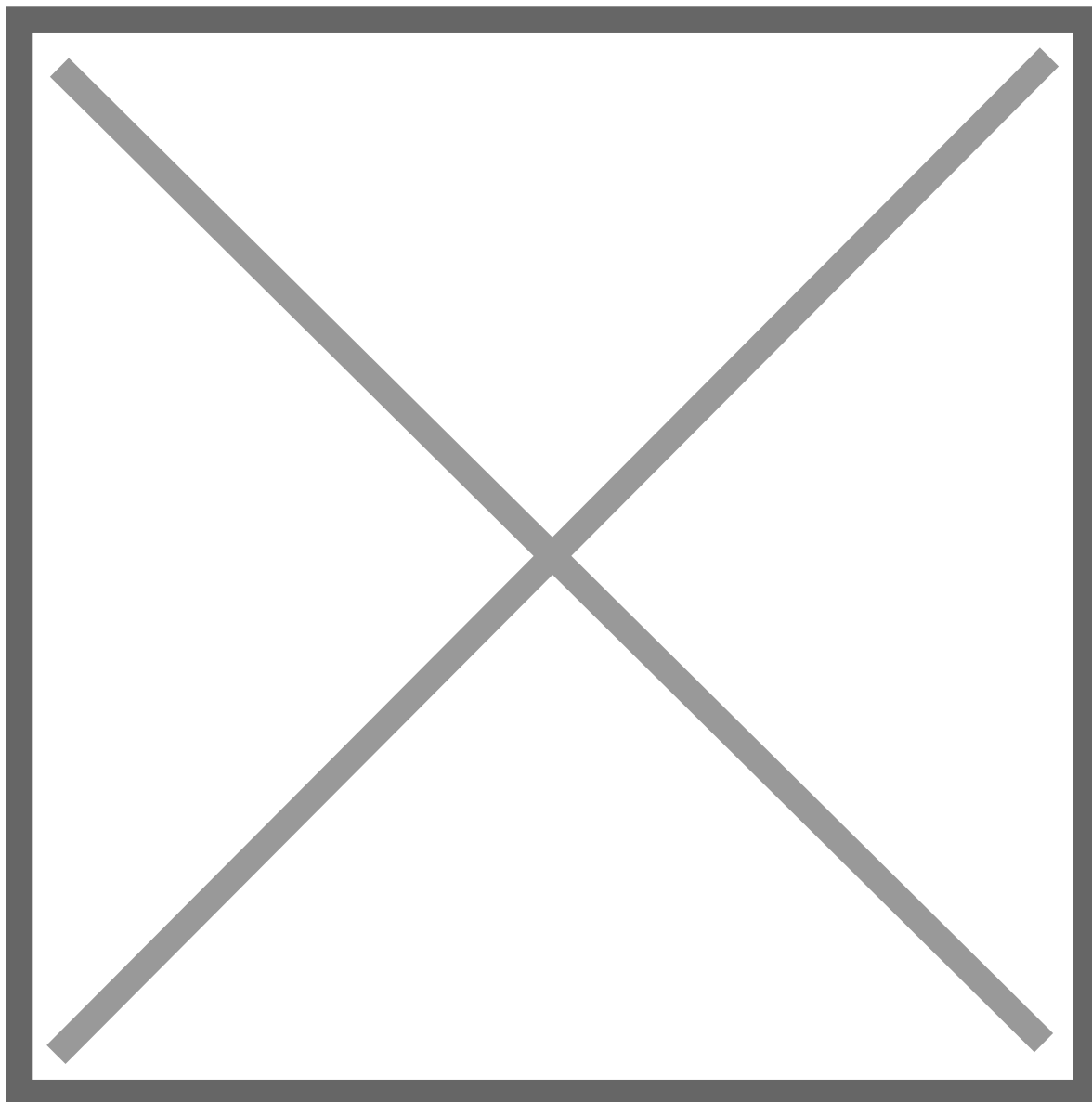
# Como Limpar o Cache no WordPress

Digamos que você tenha feito uma alteração no seu site WordPress, mas seu conteúdo não esteja sendo atualizado na interface do usuário. Isso pode acontecer porque seu servidor WordPress ainda está exibindo a versão em cache de suas páginas.

Não se preocupe: limpar o cache do WordPress geralmente resolve esse problema. Quando o cache do seu site está limpo, o sistema pode armazenar os dados atualizados e exibi-los para seus visitantes.

A maneira mais fácil de limpar o cache em sites WordPress é **instalando um plugin**. A seguir, explicaremos como limpar o cache WordPress usando quatro plugins populares ou o nosso painel de controle hPanel — além disso, mostraremos também como limpar o cache do WordPress quando seu domínio está apontado para o Cloudflare.

## Limpando Cache com o LiteSpeed



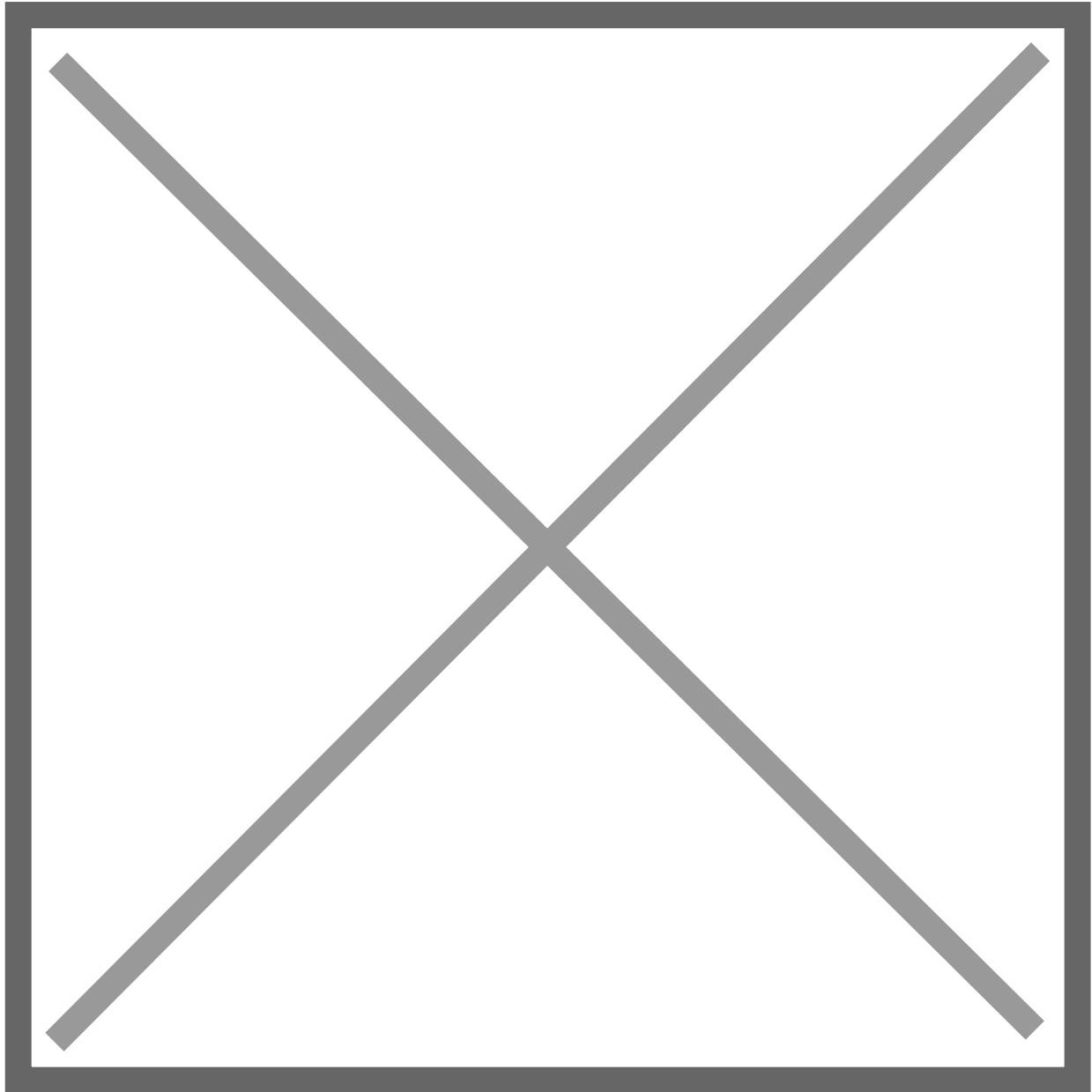
Você pode usar o plugin **LiteSpeed Cache para WordPress (LCWP)** para gerenciar de maneira eficiente a forma como o seu cache é armazenado e utilizado. Este plugin usa o servidor de cache LiteSpeed para melhorar o tempo de carregamento do site e vem com recursos exclusivos de otimização.

Todos os planos de hospedagem da Hostinger, incluindo os **planos otimizados para WordPress**, são alimentados pelo LiteSpeed — e o plugin vem pré-instalado em todas as contas de hospedagem. No entanto, se você estiver usando um provedor de hospedagem diferente, basta navegar até o diretório de plugins do WordPress para baixá-lo.

O plugin **LiteSpeed Cache** oferece várias maneiras de limpar o cache do WordPress, desde uma limpeza automática do cache com base em condições específicas até a inclusão de URLs específicas do seu site para limpeza programada. Para configurar o plugin, siga as instruções abaixo:



1. Depois de instalar e ativar o plugin, acesse seu painel de administração do WordPress. Por lá, vá para **LiteSpeed Cache**.  
limpando cache wordpress com o plugin litespeed cache
2. Escolha o menu **Toolbox**, e a janela da Caixa de Ferramentas do LiteSpeed Cache aparecerá.  
limpando cache wordpress com o plugin litespeed cache
3. Existem vários tipos de cache que você pode limpar, desde o cache de página até o cache de CSS/JS. Escolha qual tipo de limpeza de cache você deseja realizar.



4. Se você quer limpar completamente o seu cache do WordPress, escolha **Purge All**. Observe que isso não excluirá dados críticos no seu site WordPress – você pode limpar o cache sem se preocupar em causar problemas nas suas páginas.
5. No entanto, se você deseja apenas esvaziar o cache de certas páginas ou seções, role para baixo até a seção **Purge By**. A partir daí, defina a seção desejada usando uma categoria, tag ou URL.  
limpando cache wordpress com o plugin litespeed cache

Além de acessar seu painel do WordPress, você também pode limpar o cache de uma página específica simplesmente visitando-a — só tenha em mente que, antes disso, você precisa fazer login como administrador no WordPress. Em seguida, basta passar o mouse sobre o ícone LiteSpeed na sua barra de administração e escolher **Purge this page**.

limpando cache wordpress com o plugin litespeed cache

## Limpando Cache WordPress no hPanel

A segunda opção é a mais simples, mas é exclusiva para clientes da Hostinger: no hPanel, nosso painel de controle personalizado, você pode limpar o cache do seu site WordPress rapidamente e até de forma automática, se quiser. Aqui estão os passos:

1. No painel do seu site WordPress, vá até **Avançado** → **Gerenciamento de Cache**.

gerenciamento de cache no hpanel

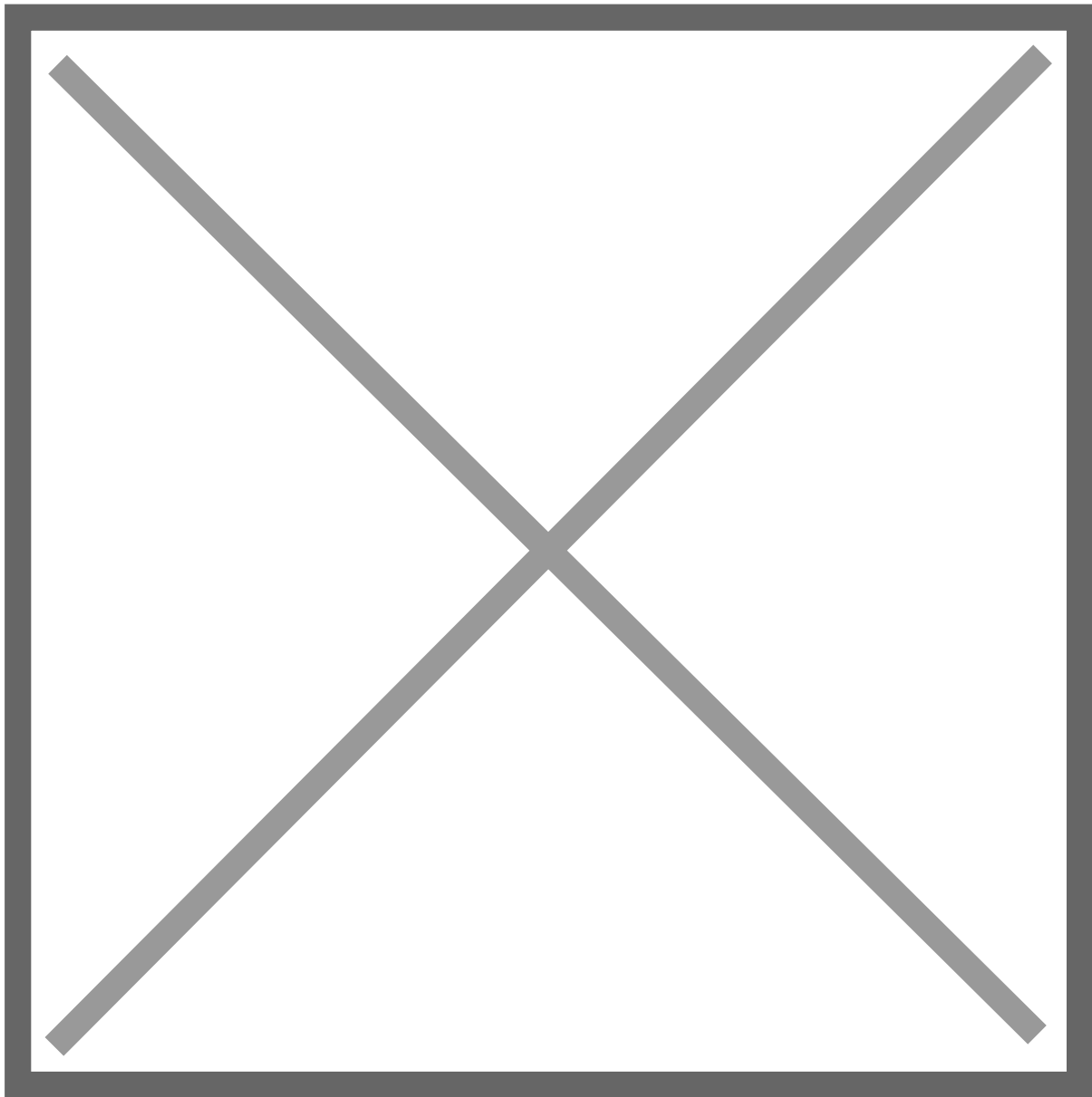
2. Ative a opção **Cache Automático** para que o sistema possa limpar o cache do site a cada 30 minutos.
3. Após ativá-la, você verá uma mensagem mostrando o status atualizado do seu cache.

status do cache atualizado no hpanel

Você também pode limpar o cache manualmente clicando no botão **Limpar Tudo**. Quando a tarefa estiver concluída, você verá uma mensagem indicando que todos os caches foram limpos.

A outra opção é limpar o cache de uma URL específica do seu site. Para fazer isso, cole a URL desejada no campo específico e clique em **Limpar**.

## Limpando Cache Com o WP Super Cache



O **WP Super Cache** modifica o sistema de cache em seu site WordPress usando três modos: **WP caching, Simple e Expert**.

Os métodos Expert e Simple carregam arquivos no modelo de super cache, então eles são boas soluções para sites com muito tráfego. A única diferença é que o método Expert reescreve regras em seu **arquivo .htaccess** para um desempenho mais rápido. Enquanto isso, o método WP caching armazena em cache apenas usuários logados, como visitantes que deixam comentários ou armazenam informações pessoais no seu site.

Este plugin também pode limpar arquivos de cache com um único clique. Simplesmente siga os passos abaixo para fazer isso:

1. Do seu painel de administração do WordPress, vá para **Configurações -> WP Super Cache**.



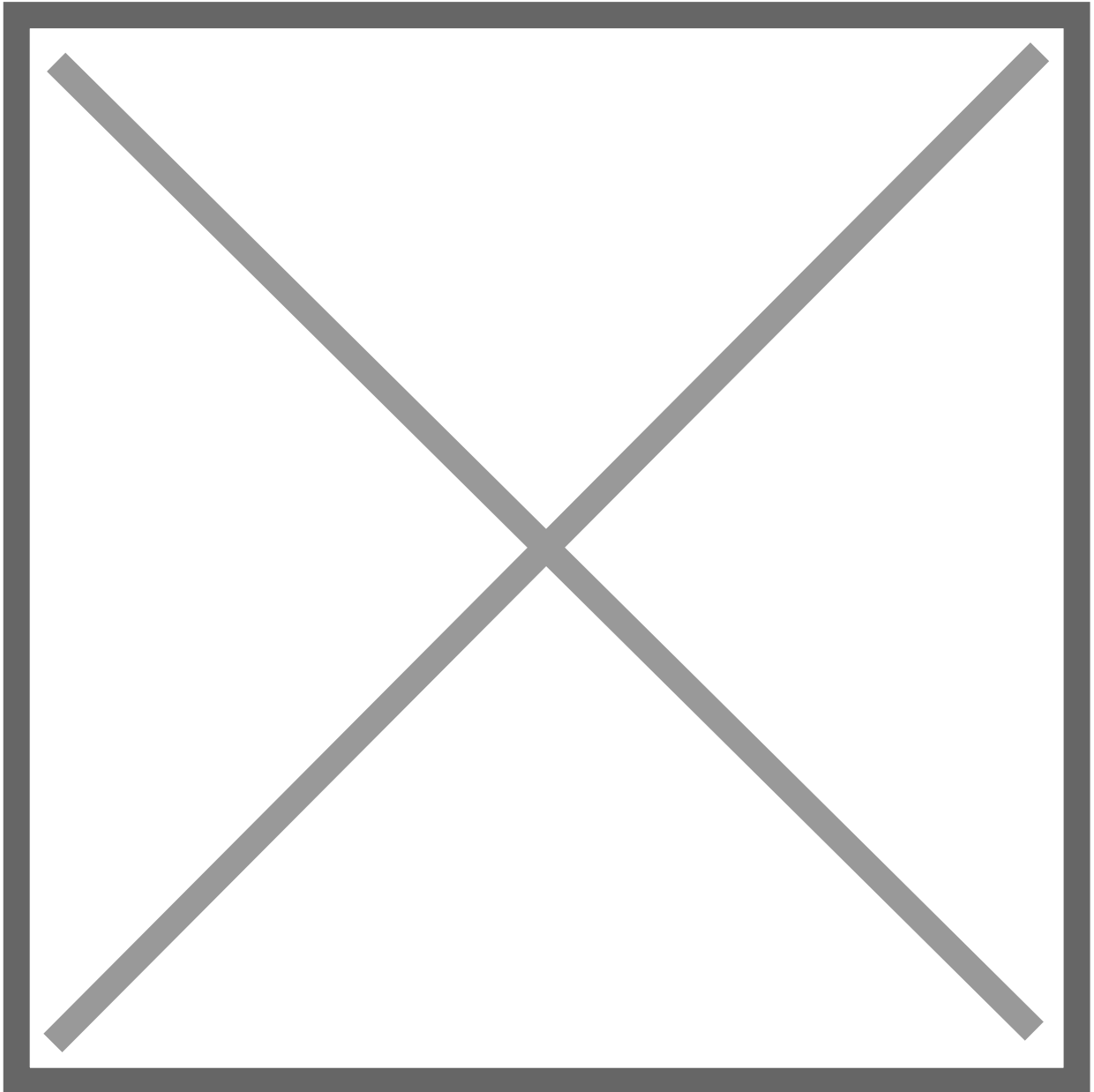
2. Na página de configurações do WP Super Cache, role para baixo até a seção **Excluir Páginas em Cache**. Então, clique no botão **Excluir Cache**.

Além disso, o plugin WP Super Cache oferece uma ampla variedade de recursos, incluindo a chamada coleta de lixo. Nele, os usuários podem definir um tempo limite para o cache, e o sistema limpará os arquivos de acordo. Para fazer isso, siga as instruções abaixo:

1. Vá até **Configurações -> WP Super Cache -> Avançado**. Em seguida, role para baixo até o **Prazo de validade e coleta de lixo**.  
limpando cache wordpress com o wp super cache
2. Defina o tempo de expiração e o intervalo do seu cache conforme sua necessidade.

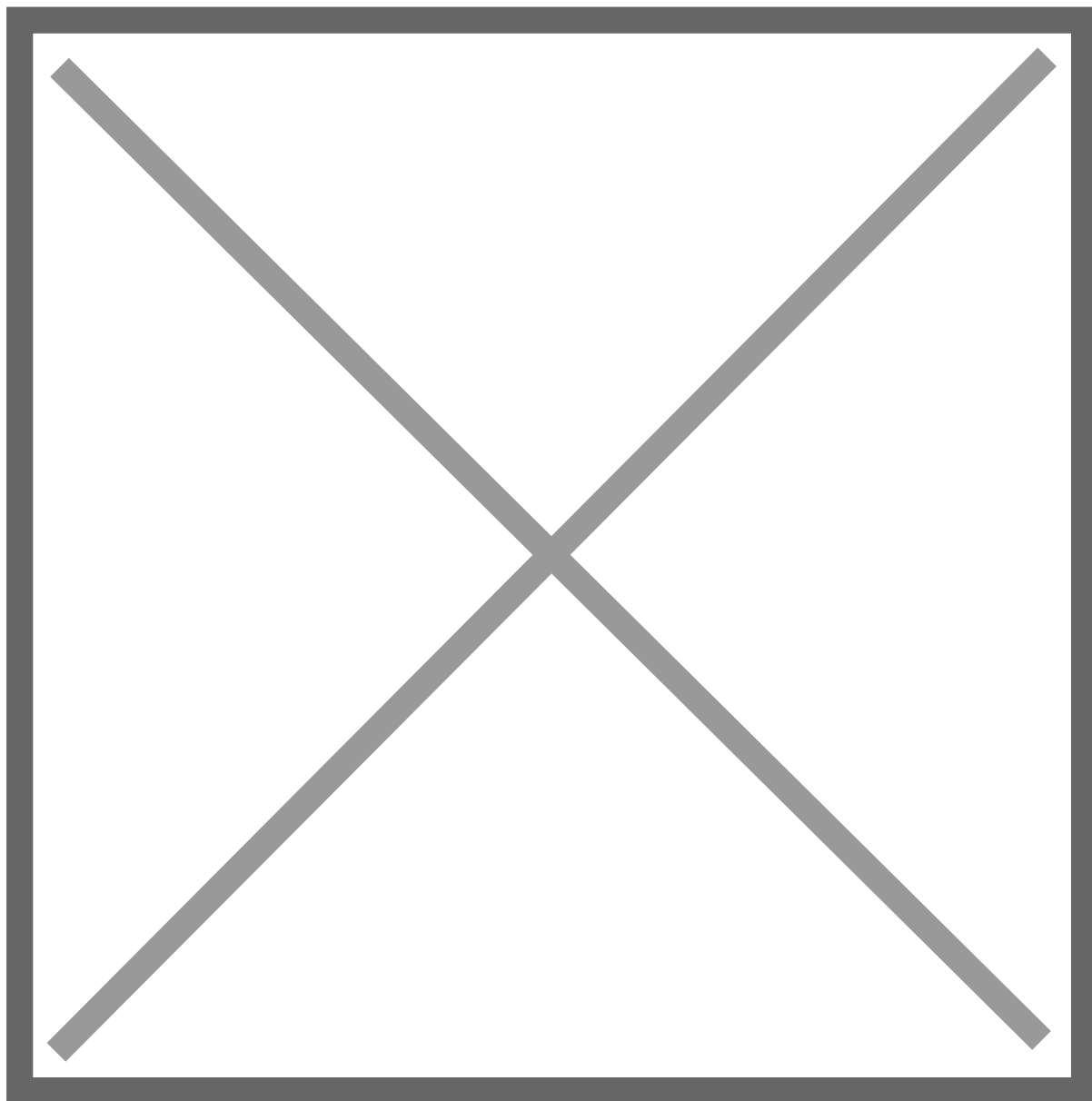
Se você não atualiza seu site WordPress com frequência, você pode definir esse prazo para **3600 segundos**, por exemplo. Por outro lado, se você administra um site movimentado com muito conteúdo dinâmico, como um site de e-commerce, é melhor definir o intervalo para **600 segundos** para evitar um desempenho muito lento.

3. Em seguida, clique em **Alterar Expiração** para salvar as alterações.



4. Se você quiser desativar a coleta de lixo, simplesmente defina o tempo de expiração do cache para **0 segundos**.

## Limpando o Cache Com W3 Total Cache

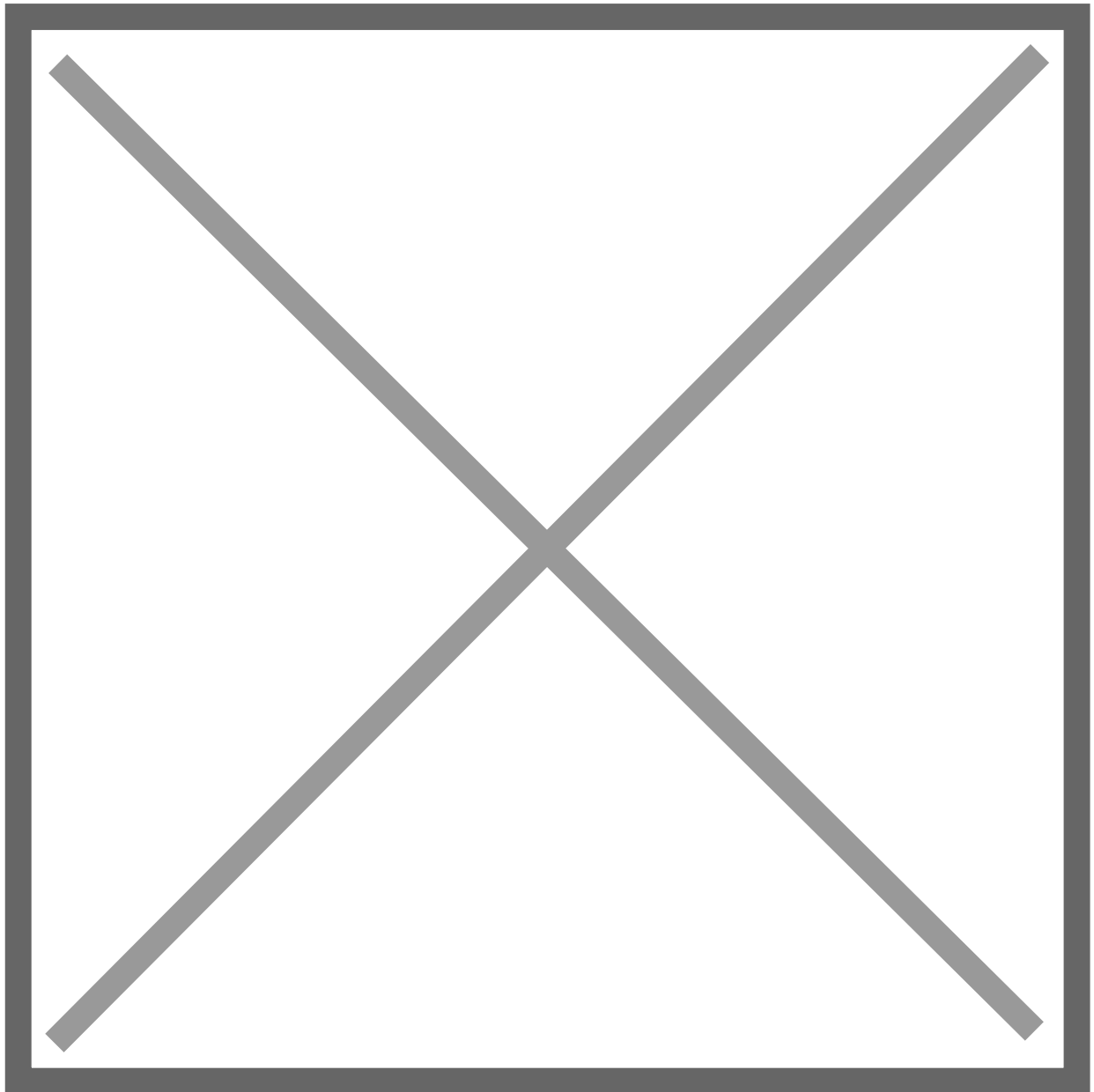


O **W3 Total Cache** é um dos plugins de cache mais populares do WordPress, conhecido por suas funcionalidades de personalização e pela economia de largura de banda. O plugin economiza até 80% do uso de largura de banda através de técnicas como minificação, otimização de feed e compressão HTTP.

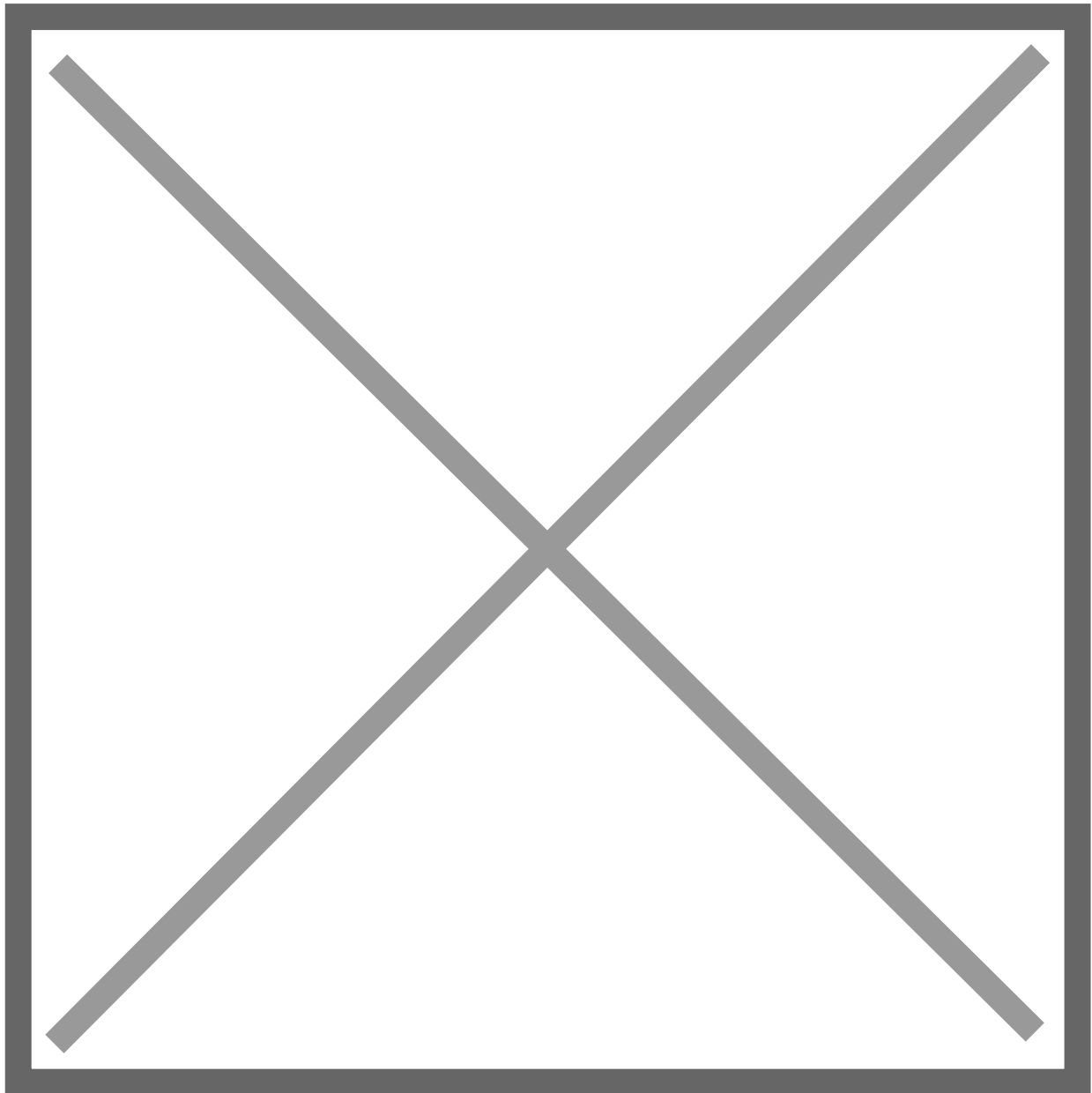
Ele oferece várias habilidades de cache, incluindo cache do navegador, banco de dados e cache do site, além de suporte AMP, que ajuda a melhorar o desempenho de SEO. Por fim, os usuários podem usar a interface de comando WP-CLI para realizar várias tarefas, como limpeza de cache e atualização de strings de consulta.

Aqui está como limpar o cache do WordPress usando o W3 Total Cache:

1. Depois de instalar o plugin, acesse sua área de administração do WordPress e vá para **Desempenho -> Painel**.

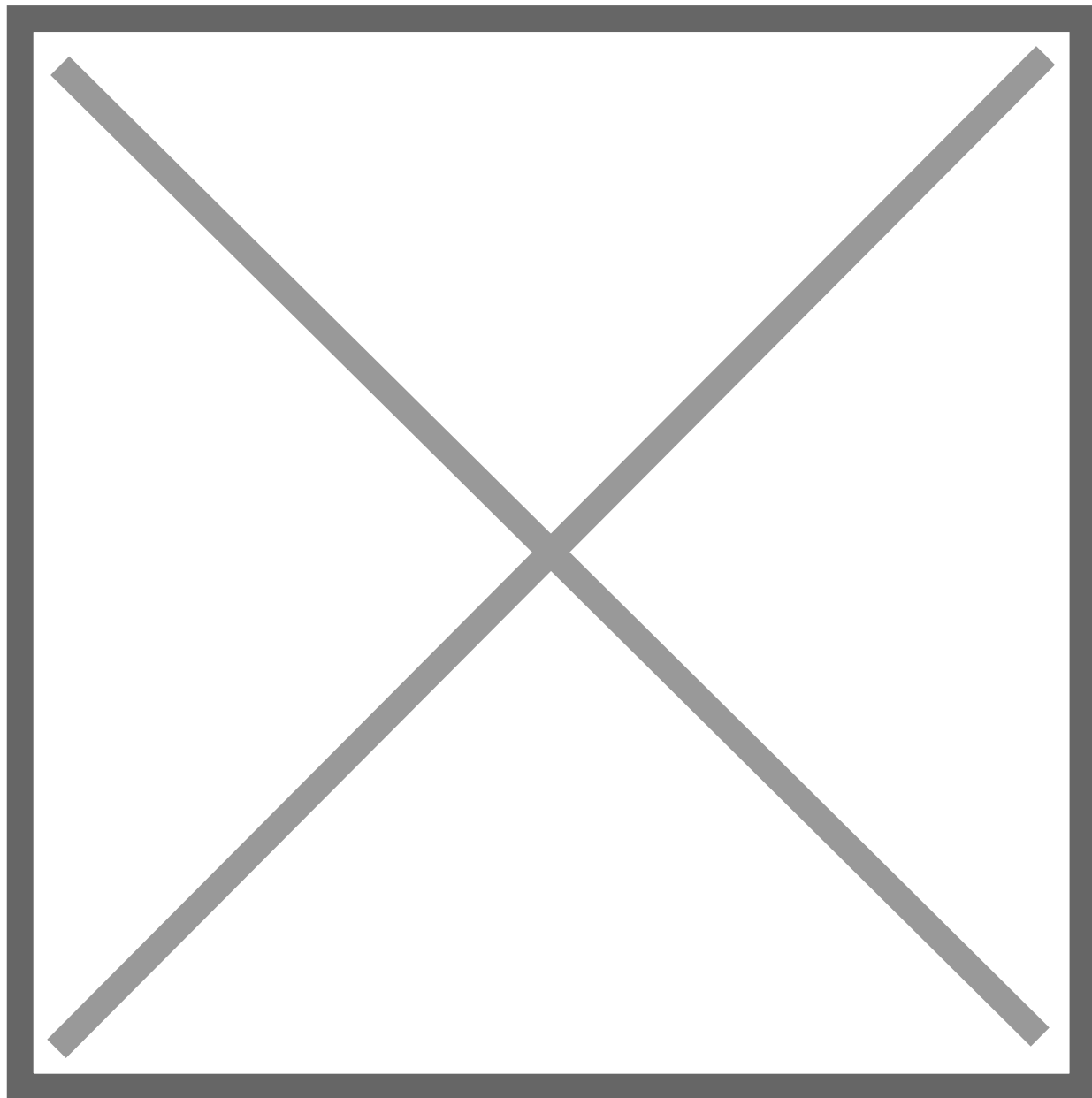


2. Clique no botão **esvaziar todos os caches** localizado no topo da página.



3. O sistema começará a limpar o seu cache do WordPress. Uma vez terminado, você deverá ver esta mensagem.  
limpando cache no wordpress com o w3 total cache
4. Além do botão **esvaziar todos os caches**, existem outros botões para deletar vários tipos de cache, incluindo a limpeza apenas do cache Memcached do banco de dados. Basta clicar no botão para iniciar a ação desejada.





# Limpando Cache Com o WP Fastest Cache

arte com recursos do plugin fastest cache para woocommerce

O **WP Fastest Cache** é outro plugin de cache popular e famoso por suas várias funcionalidades de otimização e gerenciamento de cache, como minificações, pré-carregamento de cache e **compressão GZIP**. Ele vem com um painel intuitivo e usa o módulo `mod_rewrite` para fornecer a solução de cache mais rápida.

Os usuários podem ativar o tempo limite do cache para não precisar limpar o cache manualmente. O plugin também pode deletar automaticamente todos os arquivos de cache sempre que uma

nova página ou postagem é publicada, o que impede que dados desatualizados sejam exibidos no seu site WordPress.

Siga estas etapas para realizar a limpeza automática do cache com o WP Fastest Cache:

1. Da sua área de administração do WordPress, navegue até **WP Fastest Cache**. Ele abrirá a página de opções do WP Fastest Cache.  
limpando cache de site wordpress no wp fastest cache
2. Por lá, ative a opção **Cache System**. Uma vez que você ative o armazenamento em cache, o plugin irá limpar regularmente o seu cache.  
limpando cache de site wordpress no wp fastest cache
3. Se você deseja limpar os arquivos de cache sempre que um post ou uma página for publicada, simplesmente ative as configurações **New Poste Update Post**.  
limpando cache de site wordpress no wp fastest cache
4. Para habilitar mais recursos, basta ativar as caixas de seleção desejadas. Certifique-se de clicar no botão **Submit** para salvar as alterações.

Para limpar manualmente o cache do seu site WordPress:

1. Vá para **WP Fastest Cache** e selecione a **aba Delete Cache**.
2. Por lá, clique no botão **Delete Cache**.  
limpando cache de site wordpress no wp fastest cache
3. Como alternativa, clique no botão **Delete Cache and Minified CSS/JS** se você acabou de editar arquivos CSS ou JavaScript.
4. No final da página, você encontrará as configurações de **Timeout Rules**. Basta clicar em **Add New Rule** para definir um tempo de expiração.  
limpando cache de site wordpress no wp fastest cache
5. Uma janela pop-up aparecerá. Lá, você pode configurar a condição das páginas ou posts que deseja armazenar em cache e com que frequência gostaria de excluí-los. Clique em **Salvar** para adicionar a regra.

## Limpando Cache com a Cloudflare

página inicial da cloudflare na web

Além de usar plugins, você pode usar a **Cloudflare** para limpar o cache no WordPress. A Cloudflare é uma **rede de distribuição de conteúdo (CDN)** que vem com inúmeras funcionalidades para melhorar o desempenho do seu site, incluindo controle, limpeza rápida e proteção de cache.

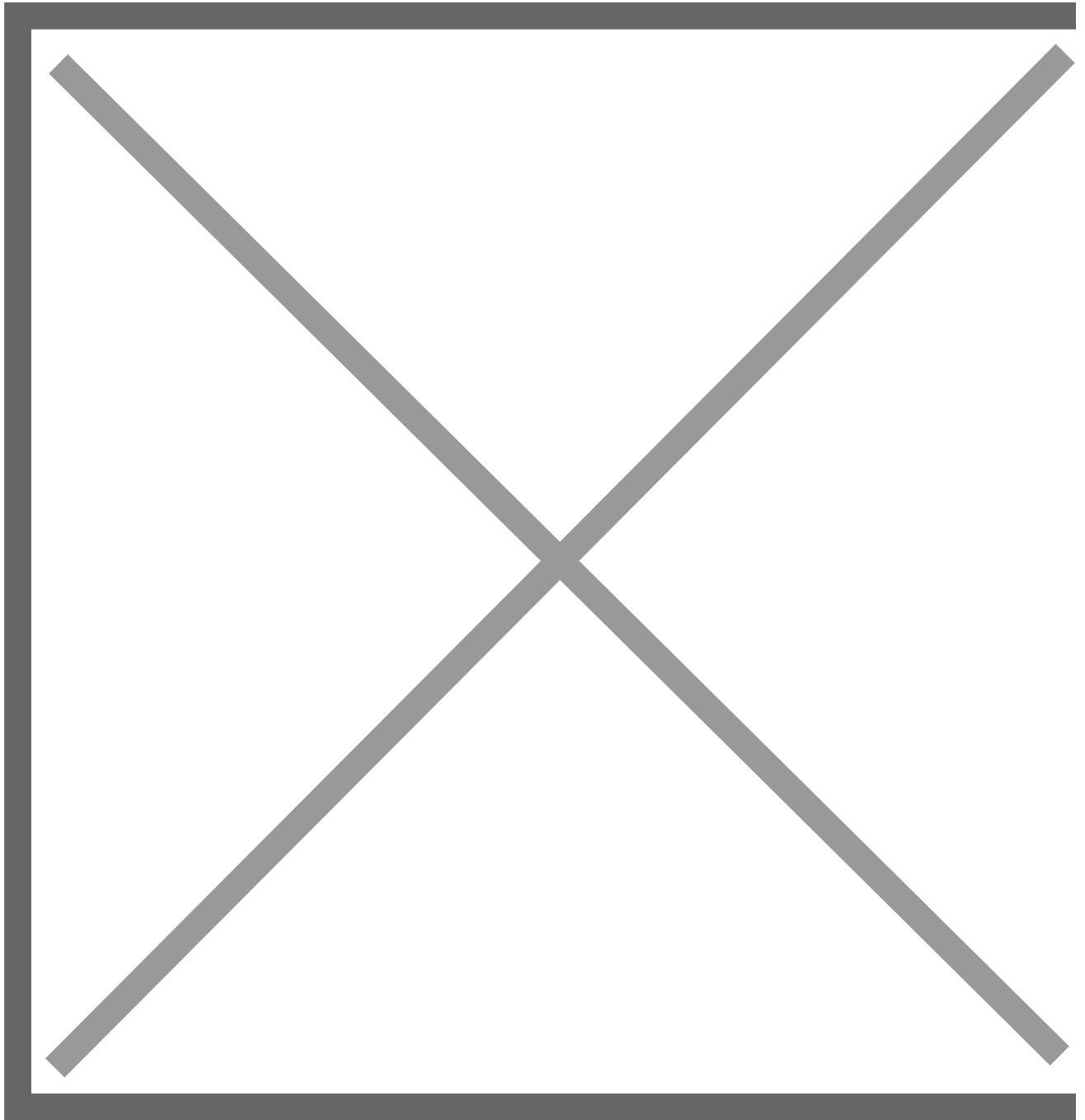
Para **configurar a CDN**, e limpar o seu cache do WordPress com este método, você deve **apontar seu domínio para a Cloudflare**. Quem tem serviços ativos da Cloudflare na Hostinger não precisa apontar seus domínios para os nameservers da Cloudflare, pois essas alterações serão feitas automaticamente.

Se sua hospedagem WordPress não configurar automaticamente a Cloudflare para você, siga os passos abaixo:

1. Crie uma **conta Cloudflare**.
2. Adicione seu domínio e aguarde o sistema identificar seus registros DNS.
3. Selecione um plano.
4. Quando a Cloudflare apresentar os nameservers necessários, simplesmente copie-os e atualize seus nameservers no painel de controle de sua conta de hospedagem.

Uma vez que seu domínio esteja apontado para a Cloudflare, siga estas etapas para limpar o cache do seu site WordPress:

1. Faça login no **painel da Cloudflare** e clique no domínio do seu site WordPress.  
limpando cache do wordpress na cloudflare
2. Navegue até o menu **cache** e clique em **Configuração**.  
limpando cache do wordpress na cloudflare
3. Escolha seu método desejado para limpar o cache. Para limpar todos os arquivos de cache, clique no botão **Purge Everything** . Se você deseja selecionar páginas específicas, escolha **Custom Purge** e preencha as páginas que deseja limpar.  
limpando cache do wordpress na cloudflare



# Limpendo Cache no Navegador

Limpar o cache do seu próprio navegador também ajuda a solucionar problemas de carregamento e formatação. Isso removerá arquivos desatualizados, permitindo que o browser carregue a versão atualizada do site desejado. Fazer isso também pode resolver problemas relacionados ao servidor, incluindo o **502 bad gateway** e o **504 gateway timeout**.

Lembre-se de que limpar o cache do navegador também pode excluir alguns dados de navegação, como logins salvos preferências de configuração. Portanto, certifique-se de **exportar seus dados**

do navegador e **importá-los de volta** uma vez que você terminar de limpar o cache.

As seguintes etapas mostrarão como limpar o cache no Google Chrome. Observe que os mesmos passos se aplicam a outros navegadores também.

1. Abra o navegador Chrome.
2. Clique no menu de três pontos ( **:** ) no canto superior direito da sua tela.
3. Escolha **Mais ferramentas -> Limpar dados de navegação** .
4. Isso te levará a uma página onde você pode realizar a limpeza do cache de forma Básica ou Avançada.

limpando dados de navegação no chrome

5. Escolha seu método desejado, defina o intervalo de tempo e marque as caixas conforme necessário. Deixe a opção **Senhas e outros dados de login** desmarcada para manter suas credenciais de login.
6. Finalmente, clique no botão **Limpar dados**.

## Conclusão

O armazenamento em cache é uma ótima maneira de fazer seu site carregar mais rápido, pois ele armazena cópias do conteúdo estático do seu site para reutilização futura. No entanto, arquivos de cache desatualizados podem impedir que os visitantes vejam novas alterações no seu site.

Felizmente, limpar o cache do seu site pode resolver esse problema. Este tutorial mostrou como limpar o cache no WordPress usando alguns plugins de cache populares ou o hPanel. Se a remoção do cache do servidor WordPress não resolver o problema, tente também **limpar o cache do seu navegador**.

Observe que, depois que os arquivos em cache sejam removidos, os navegadores e servidores precisarão de algum tempo para baixá-los novamente — o que pode resultar em um tempo de carregamento um pouco mais longo.

Esperamos que este artigo tenha sido útil. Se você tiver alguma dúvida sobre o cache do WordPress, sinta-se à vontade para deixar um comentário logo abaixo. Boa sorte!

## Mais Tutoriais do WordPress

**O Que é WordPress**

**Como Fazer Backup de um Site WordPress**

**Removendo Malware no WordPress**

[Como Otimizar Imagens no WordPress](#)

[Como Instalar o WordPress](#)

[Guia de Segurança do WordPress](#)

# Como Limpar o Cache WordPress: Perguntas Frequentes

Agora que você já sabe como limpar o cache do WordPress, aqui estão algumas perguntas frequentes feitas sobre o tema.

## Com Que Frequência Devo Limpar o Cache no WordPress?

A frequência para limpar o cache no WordPress depende de quão frequentemente você atualiza o seu site. Recomendamos a limpeza do cache pelo menos uma vez por semana para mantê-lo atualizado e seguro. No entanto, você também pode limpar o cache todos os dias, se fizer atualizações de hora em hora ou diárias em seu site.

## Como Limpar o Cache Sem um Plugin do WordPress?

Limpar o cache sem um plugin WordPress é possível — você pode fazer a tarefa manualmente. Para fazer isso, você precisa deletar o conteúdo da pasta wp-content/cache do seu site usando um cliente FTP, como o FileZilla, ou acessando seu servidor diretamente.

## Quais São os Benefícios de Limpar o Cache?

Limpar o cache é uma prática de manutenção essencial para qualquer usuário da web. Ela ajuda a liberar espaço no seu dispositivo e manter o desempenho do navegador ao mesmo tempo que permite o acesso ao conteúdo mais recente dos sites que você visita com frequência.

# Compressão GZIP para melhor desempenho do Wordpress

Link: <https://www.hostinger.com.br/tutoriais/melhorando-o-desempenho-do-site-compressao-gzip>

## Tutorial: Como Ativar a Compressão GZIP Para Acelerar Seu Site (hostinger.com.br)

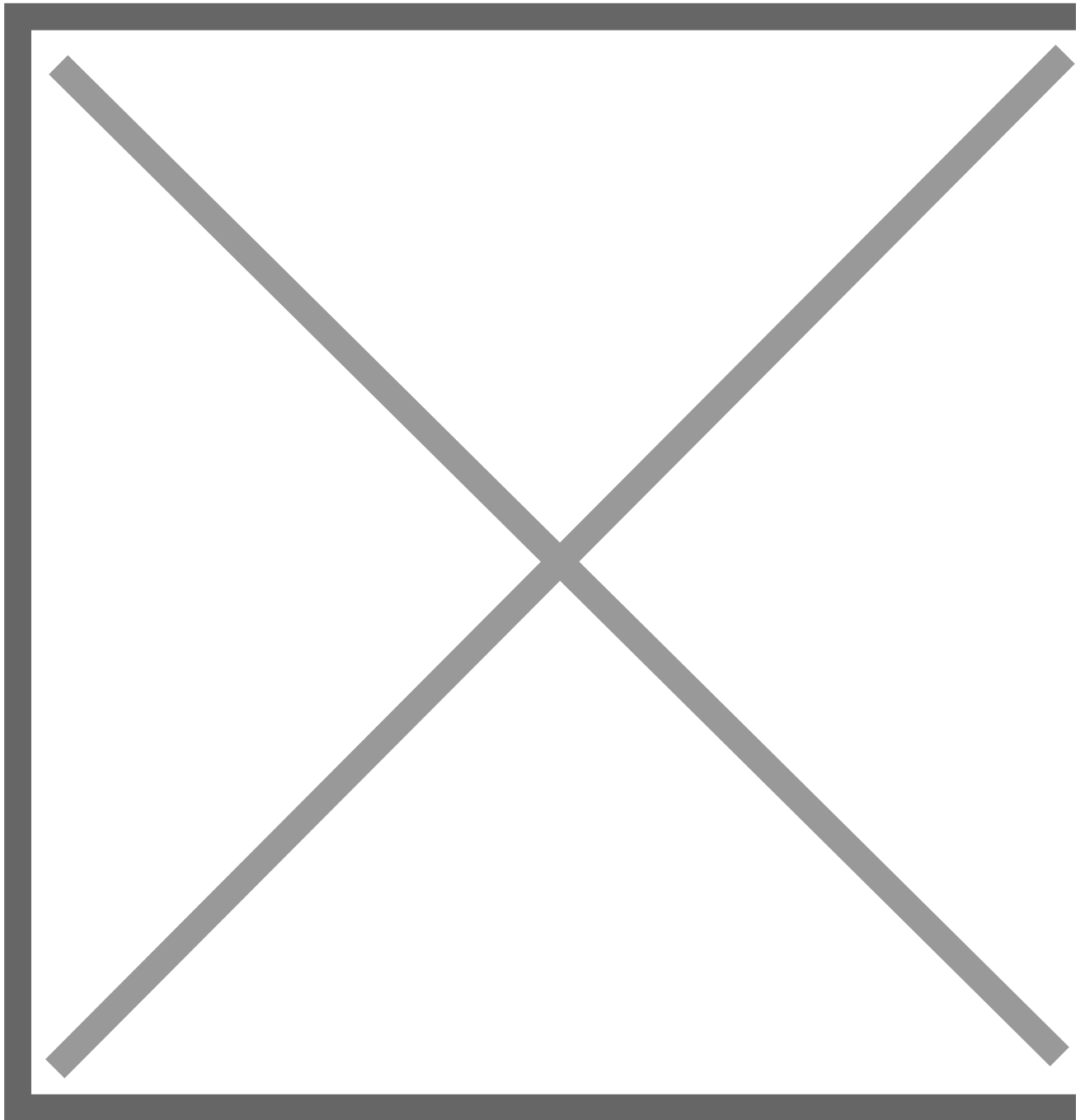
A otimização para mecanismos de busca (**SEO**) é uma parte importantíssima da criação de conteúdo digital. Afinal, o ranqueamento do seu site e das suas páginas nos resultados dos serviços de busca impacta diretamente sua visibilidade e tráfego.

Um dos fatores decisivos do ranking de SEO é a velocidade do site. Portanto, é muito importante garantir que seu site seja rápido e tenha excelente desempenho. Caso contrário, seus visitantes podem se frustrar e clicar no botão de “fechar” antes mesmo da página terminar de carregar.

É aqui que a compressão Gzip vem a calhar: ela representa um dos truques mais eficientes para que você melhore a velocidade das suas páginas.

Ao final deste artigo, você saberá o que é compressão Gzip e como ativá-la no seu site. Vamos lá?





## Conteúdo

- O Que É Compressão Gzip
- Como Ativar a Compressão Gzip
  - Verifique a Compressão Gzip Usando o Teste de Velocidade do Site
  - Verifique a Compressão Gzip Usando um Teste Específico
  - Verifique a Compressão Gzip Usando as Ferramentas de Desenvolvedor do Navegador

- [Ative a Compressão Gzip Usando um Plugin do WordPress](#)
- [Ative a Compressão Gzip Usando o Arquivo .htaccess](#)
- [Como Funciona a Compressão Gzip](#)
- [Por que Ativar a Compressão Gzip](#)

# O Que É Compressão Gzip

Quanto maior o arquivo, mais tempo ele precisa para ser baixado, aumentando o tempo de carregamento do seu site. Felizmente, as tecnologias de compressão de dados permitem que os arquivos tenham seus tamanhos reduzidos. Com isso, seu site carregará mais rápido.

Existem dois tipos principais de compressão:

- **Compressão com perdas** – elimina algumas informações do arquivo original.
- **Compressão sem perdas** – preserva todos os dados do arquivo original.

A compressão Gzip se enquadra na categoria de compressão sem perda. Por isso, trata-se de um método popular para reduzir o tamanho de arquivos HTML, **PHP**, **JavaScript**, **CSS** e outros.

## Como Ativar a Compressão Gzip

Antes de habilitar a compressão Gzip, primeiro verifique se o seu site já tem isso habilitado.

A seção a seguir mostrará como fazer essa verificação usando testes de velocidade do site, testes de compressão Gzip e ferramentas de desenvolvedor do navegador.

## Verifique a Compressão Gzip Usando o Teste de Velocidade do Site

Um **teste de velocidade de site** é uma ferramenta para medir o tempo que um site leva para carregar. Ele indica o tempo de carregamento do site ou página e, em muitos casos, fornece um relatório detalhado dos elementos que afetam o desempenho. Algumas ferramentas exibirão até mesmo a taxa de redução do tamanho dos arquivos do site.

Esses testes também sugerem inúmeras dicas e truques para melhorar a velocidade do site, incluindo recomendações de algoritmos de compressão para produzir arquivos menores.

Ferramentas de testes gratuitas estão disponíveis em muitos sites, incluindo **Pingdom**, **PageSpeed Insights** do Google e **GTmetrix**.

## Verifique a Compressão Gzip Usando um Teste Específico

Você pode usar uma ferramenta gratuita para verificar se o seu site está com a compressão Gzip ativada. Algumas opções incluem a **Verificação de Compressão Gzip** do Small SEO Tools e o **Teste de Compressão HTTP** do WhatsMyIP.

Basta abrir a ferramenta, colar a URL do site no campo fornecido e clicar em **Verificar Compressão** ou algo do tipo.

## Verifique a Compressão Gzip Usando as Ferramentas de Desenvolvedor do Navegador

Este método é mais indicado para quem não quer usar uma ferramenta externa para verificar a compressão Gzip de um site.

Siga as etapas abaixo para verificar se o Gzip está ativado em um site usando um navegador:

1. Digite o URL do site no navegador.
2. Abra as ferramentas de desenvolvedor do navegador ou clique com o botão direito na página e selecione **Inspecionar**.
3. Clique na aba **Rede**. Você deverá ver todos os elementos que o servidor web enviou. Caso não, recarregue a página.
4. Selecione um dos elementos para visualizar seu conteúdo. Clique na aba **Cabeçalhos**.
5. Em **Cabeçalhos de Resposta**, encontre **content-encoding** – o método de compressão usado pelo site será listado ao lado dele.

Se o Gzip não está ativado, existem duas maneiras de fazer isso. Você pode modificar o arquivo **.htaccess** ou optar por um **plugin do WordPress**.

# Ative a Compressão Gzip Usando um Plugin do WordPress

A maneira mais fácil de ativar o Gzip em um site WordPress é **instalar um plugin**. Esse caminho é muito mais fácil e seguro do que modificar o arquivo **.htaccess**.

Existem muitos plugins disponíveis para esse propósito, mas escolhemos esses quatro plugins com base em suas funcionalidades e facilidade de uso:

- **WP-Optimize Cache**. Uma solução completa para impulsionar o desempenho do site. O plugin habilita Gzip e **cache da web**, entre outros recursos. Além disso, ele pode eliminar arquivos desnecessários de um site. Esta ferramenta pode até otimizar imagens para liberar mais espaço.

plugin wp-optimize parra compressão gzip

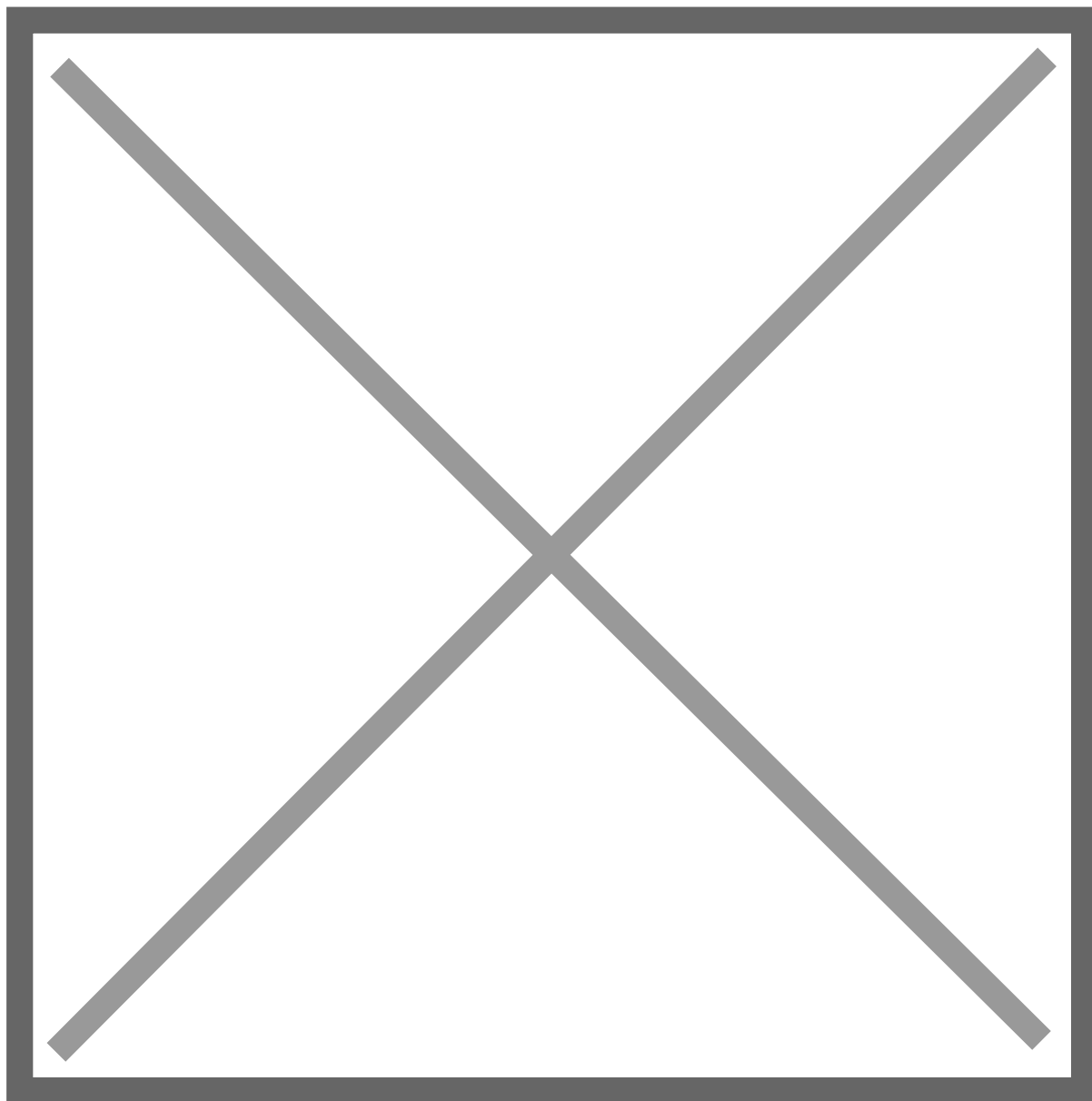
- **Hummingbird**. Este plugin dá aos usuários mais liberdade para escolher quais aspectos otimizar. A compressão Gzip, a minificação e o lazy loading de imagens são algumas de suas principais características.

plugin hummingbird para wordpress

- **WP Fastest Cache**. Esta ferramenta combina recursos de cache com Gzip para tornar suas páginas rápidas e leves. Este plugin também pode desativar automaticamente emojis e otimizar cada postagem individualmente.

arte com recursos do plugin fastest cache para woocommerce

- **WP Rocket**. Um plugin de desempenho equipado com inúmeros recursos, como cache de página e navegador, pré-carregamento de cache, compressão Gzip, otimização para comércio eletrônico e excelente compatibilidade com plugins populares do WordPress, temas e tipos de hospedagem.



Como esses plugins têm mais recursos do que apenas compressão, sugerimos que você experimente todos eles para ver qual se adapta melhor às suas necessidades.

## Ative a Compressão Gzip Usando o Arquivo .htaccess

Essa forma de habilitar a compressão Gzip é mais técnica, portanto mais adequada para usuários avançados. No entanto, qualquer pessoa pode seguir com os passos abaixo, desde que siga os cuidados necessários.

**Importante!** Antes de continuarmos, faça um backup do arquivo .htaccess caso algo dê errado.

Este tutorial mostrará como modificar o arquivo **.htaccess** usando o **Gerenciador de Arquivos** do painel de controle do seu provedor de **hospedagem**. Neste caso, vamos usar o hPanel da Hostinger. No entanto, você também pode usar um cliente FTP, como o **FileZilla**.

Para habilitar a compressão através do arquivo **.htaccess**, siga os passos abaixo:

1. Faça login no hPanel e clique em **Sites** na barra de navegação. Então, encontre o site desejado e clique em **Gerenciar**.

botão gerenciar na tela sites do hpanel

2. Role para baixo para encontrar a seção **Arquivos** e clique em **Gerenciador de Arquivos**.

gerenciador de arquivos do hpanel

3. Dê um duplo clique na pasta **public\_html**.

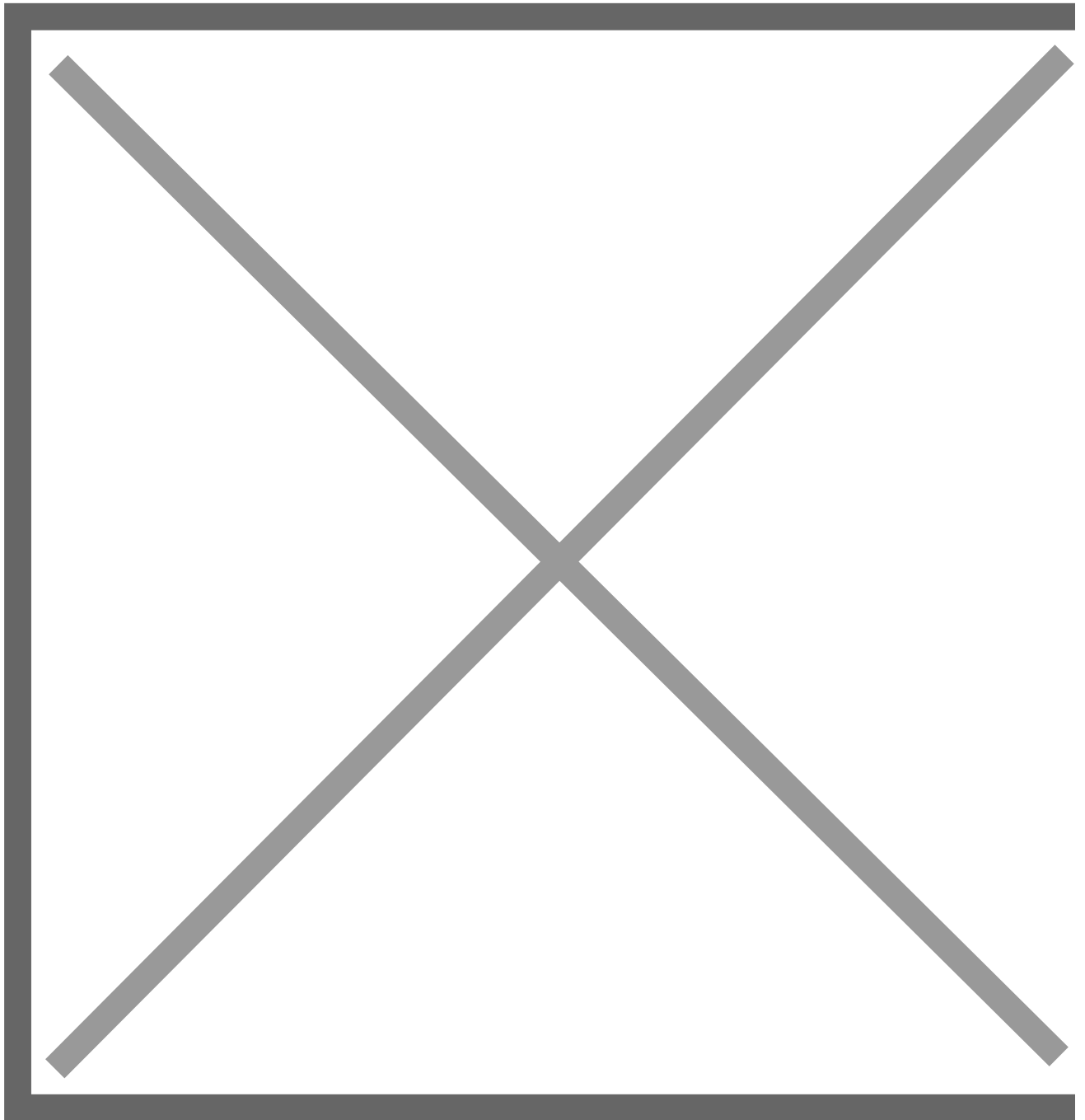
pasta public\_html em site no gerenciador de arquivos da hostinger

4. Encontre o arquivo **.htaccess**. Baixe o arquivo e salve uma cópia no seu computador como backup. Então, dê um duplo clique no arquivo **.htaccess** localizado na pasta **public\_html** para abri-lo.
5. Cole o código abaixo antes da linha **# END WordPress**.

```
<IfModule mod_deflate.c>
AddOutputFilterByType DEFLATE text/plain
AddOutputFilterByType DEFLATE text/html
AddOutputFilterByType DEFLATE text/xml
AddOutputFilterByType DEFLATE text/css
AddOutputFilterByType DEFLATE application/xml
AddOutputFilterByType DEFLATE application/xhtml+xml
AddOutputFilterByType DEFLATE application/rss+xml
AddOutputFilterByType DEFLATE application/javascript
AddOutputFilterByType DEFLATE application/x-javascript
</IfModule>
```

6. Uma vez terminado, clique no botão **Salvar**.

Se você quer ter certeza de que o Gzip está ativado, use um dos métodos que explicamos anteriormente.



# Como Funciona a Compressão Gzip

Para entender o Gzip, você primeiro precisa saber como um site funciona.

Um site é composto por arquivos armazenados em um servidor web. Quando uma pessoa digita a URL do site em um navegador, o navegador envia uma solicitação para o servidor de hospedagem de site que contém o conteúdo desejado.

infográfico sobre como funciona a compressão gzip

O servidor web processa a solicitação recuperando os arquivos solicitados de um banco de dados. Ele envia os arquivos para o navegador na forma de uma resposta HTTP. Naturalmente, quanto maior o tamanho, mais tempo a transferência de dados levará.

Uma vez que o navegador recebe os arquivos, ele irá renderizar a página da web em questão.

No entanto, normalmente haverá dados repetidos dentro de arquivos de texto, como caracteres ou palavras duplicadas aparecendo várias vezes ao longo de um documento.

Dentro de arquivos HTML, por exemplo, haverá uma tag de fechamento semelhante para cada tag de abertura. Por exemplo, a tag de abertura **<table>** será combinada com **</table>**, enquanto a tag **<div>** será combinada com **</div>**.

O Gzip emprega o algoritmo **DEFLATE**, sem perdas, que encontra e substitui essas strings duplicadas por identificadores únicos.

Esse processo é chamado de codificação de caracteres. Ele produz arquivos compactados ou comprimidos cujos nomes terminarão com a extensão de arquivo **.gz**. O arquivo no tamanho comprimido é menor comparado à versão original ou não comprimida. Isso economiza largura de banda e resulta em transferências de dados mais rápidas.

Assim que o navegador do usuário recebe o arquivo compactado, ele irá descompactar ou descomprimir o arquivo. Os identificadores únicos no arquivo compactado serão decodificados de volta para os dados originais.

**Importante!** Embora o Gzip possa ser usado para processar arquivos de qualquer tipo, não haverá uma redução significativa no tamanho do arquivo para tipos de arquivos já compactados, como imagens ou vídeos. Tais arquivos podem até aumentar de tamanho.

Além disso, note que Gzip não é o mesmo que ZIP. Este último também é um método de compressão sem perdas, mas é usado para armazenar arquivos em vez de transferi-los. Ele produz um arquivo **.zip** em vez de um arquivo **.gz**.

# Por que Ativar a Compressão Gzip

O Gzip é apenas um dos muitos métodos e algoritmos de compressão de dados. Exemplos de outros métodos incluem ZIP, TAR e 7Z.



Você pode se perguntar por que deve ativar a compressão Gzip em vez de outros métodos, especialmente quando eles supostamente têm um desempenho melhor. Por exemplo, **pesquisas conduzidas pela Expedited Security** mostram que o algoritmo de compressão Brotli, do Google, pode comprimir arquivos HTML, CSS e JavaScript em tamanhos menores que o Gzip.

Dito isso, existem muitos motivos para escolher o Gzip em vez de outros métodos ou algoritmos, incluindo:

- **Livre e de código aberto.** O software Gzip é gratuito para usar. Ele foi criado inicialmente como um substituto gratuito para o **compress**, um programa de compressão de shell Unix patenteado.
- **Grande popularidade.** De acordo com dados de junho de 2022, cerca de **60%** de todos os sites comprimem arquivos com o método Gzip.
- **Exigência mínima de recursos.** O Gzip usa relativamente poucos recursos de processamento e armazenamento de dados temporários em comparação com métodos semelhantes.
- **Taxa efetiva de compressão.** O método Gzip pode reduzir o tamanho de arquivos de dados em cerca de **70%** (arquivos pequenos) ou até **90%** (arquivos maiores).
- **Velocidade de compressão e descompressão.** O Gzip é mais rápido do que muitos de seus concorrentes, como Brotli e BZIP2.

Com seu tempo de compressão rápido e eficiente, o Gzip ajuda a reduzir o tempo de carregamento da página, aumentando a sua classificação de SEO.

Velocidades de carregamento mais rápidas impactam positivamente o ranqueamento das páginas nos resultados das páginas de motores de busca. Quanto mais rápido o site carrega, melhor ele se classifica. Quanto maior a classificação, maior a visibilidade que você recebe, **direcionando tráfego para o site** e aumentando sua receita.

## Conclusão

O Gzip é um formato de arquivo e um tipo de tecnologia de compressão sem perdas usado para reduzir o tamanho dos arquivos e acelerar sites. Ele funciona usando o algoritmo **DEFLATE**, que encontra strings duplicadas dentro de um arquivo de texto e as substitui por identificadores únicos. Eles são decodificados de volta aos dados originais durante o processo de descompressão.

Após habilitar a compressão Gzip, os visitantes do seu site terão uma experiência de usuário aprimorada, e possivelmente ele terá uma classificação mais alta nas páginas de resultados dos motores de busca.

Você pode verificar se um site já ativou o Gzip usando um teste de velocidade online ou as ferramentas de desenvolvedor do seu navegador.

A compressão Gzip em um site pode ser ativada usando dois métodos – instalando um plugin do WordPress ou editando o **arquivo .htaccess**. Recomendamos a instalação de um plugin em vez de editar o arquivo **.htaccess** para iniciantes: é um método mais fácil e menos arriscado.

Embora diversos métodos e algoritmos de compressão estejam disponíveis, existem muitos motivos para escolher a compressão Gzip em detrimento de outros. Não apenas ele é gratuito e de código aberto, mas também possui uma das taxas de compressão mais eficazes e é muito rápido.

Esperamos que este artigo tenha ajudado você a entender melhor a compressão Gzip. Desejamos boa sorte na melhoria do seu site!

## Aprenda Sobre Outros Métodos de Otimização de Sites

**Como Ativar o Keep-Alive**

**Como Minificar Arquivos CSS, HTML e JS**

**Otimização de Site: Top 10 Estratégias para Melhorar Velocidade, UX, SEO**

# Como Otimizar Seu Site WordPress Em 11 Passos – Guia 2024

Link: <https://www.hostinger.com.br/tutoriais/acelerar-seu-site-wordpress-em-9-passos-faceis>

Ter um site otimizado é crucial para qualquer empresa ou projeto.

Um site lento não apenas deixa uma impressão ruim nos visitantes, como também aumenta a chance deles deixarem o seu site. Uma página que leva cinco segundos para carregar **tem uma chance 90% maior** de sofrer rejeição do que uma que leva um segundo.

Além disso, sites de busca como o Google consideram a **velocidade de carregamento como um fator** na hora de determinar seus rankings. Sites lentos têm maior dificuldade de serem descobertos de forma orgânica.

Felizmente, existem maneiras de consertar um site lento. Neste guia, vamos explorar 11 passos que você pode executar agora, e que vão acelerar sites do WordPress de maneira significativa. Nós vamos trazer insights de especialistas que vão ensinar dicas valiosas e trazer um conhecimento mais profundo de otimização do WordPress.

## Conteúdo

- [Tutorial em Vídeo](#)
- [Escolha um Plano de Hospedagem Adequado para seu Site do WordPress](#)
- [Escolha o Servidor Localizado Mais Próximo do seu Público-Alvo](#)
- [Use uma Rede de Fornecimento de Conteúdo](#)
- [Entregue o Conteúdo Através do HTTP/2](#)
- [Escolha um Tema WordPress Otimizado](#)
- [Otimize as Imagens](#)
- [Atualize Regularmente o Núcleo, os Plugins e os Temas do WordPress](#)
- [Desinstale Plugins Desativados](#)

- [Minifique o HTML, o CSS e o JavaScript](#)
- [Use a Versão PHP Mais Recente](#)
- [Otimizar Banco de Dados](#)
- [Recomendações de Leitura](#)
- [Crie Seu Site WordPress Para o Futuro](#)

## Tutorial em Vídeo

<https://www.youtube.com/embed/JRGQytM8Sa0?feature=oembed>

## Escolha um Plano de Hospedagem Adequado para seu Site do WordPress

Primeiramente, certifique-se que você usa um provedor de hospedagem confiável e tem um plano que se encaixa com os requisitos do seu site. Uma hospedagem de baixa qualidade e um plano inadequado podem deixar seu site lento, além de causar outros problemas.

A chave é fazer uma boa pesquisa na hora de escolher sua hospedagem e o seu plano. Identifique as necessidades do seu site e então encontre um provedor de hospedagem e um plano que consiga satisfazê-las. Caso escolha a Hostinger, você mesmo pode dar início à [migração do seu site](#) para os nossos serviços – e gratuitamente.

Confirme que a provedora de hospedagem que você escolher tem um histórico comprovado de oferecer excelente desempenho com o WordPress e um ótimo suporte ao cliente. Algumas empresas até oferecem recursos únicos que deixam o WordPress mais rápido.

A Hostinger, por exemplo, tem um recurso de gestão de cache no hPanel, e usa o **LiteSpeed** para otimizar o desempenho do servidor e os tempos de carregamento.

A arquitetura assíncrona do LiteSpeed consegue lidar com maior tráfego e com conexões simultâneas melhor do que servidores que usam o Apache. Ao invés de criar uma nova solicitação toda vez, o LiteSpeed armazena arquivos de cache compactados, o que otimiza a velocidade da página.

O LiteSpeed também vem com um plugin de cache para o WordPress que pode acelerar ainda mais o seu site. Ao contrário de outros plugins de cache, o LiteSpeed se integra diretamente com o servidor, permitindo que ele se lembre das entradas e analise as dependências para uma gestão de cache mais inteligente.

Uma vez que você tiver escolhido um provedor de hospedagem, é hora de escolher o plano mais adequado para o seu site do WordPress. Em geral, os provedores oferecem uma grande **[variedade](#)**

**de tipos de hospedagem**, como por exemplo:

- **Hospedagem de site**. Um tipo de hospedagem de entrada que armazena diferentes sites no mesmo servidor, com compartilhamento de recursos. Os planos de hospedagem compartilhada da Hostinger partem de **R\$ 6.99/mês**.
- **Servidor Cloud**. Um tipo de hospedagem premium que oferece os recursos aprimorados de uma **rede com múltiplos servidores**. É tão simples de usar quanto a hospedagem compartilhada. A linha de planos de hospedagem Cloud da Hostinger parte de **R\$ 39.99/mês**.
- **Servidor VPS Brasil**. Um tipo de hospedagem que dá acesso root e recursos dedicados para os usuários. Ela requer algum conhecimento técnico para ser configurada e mantida. A hospedagem VPS KVM da Hostinger custa a partir de **R\$ 29.99/mês** e conta com servidores dedicados no Brasil para máximo desempenho.
- **Hospedagem WordPress**. Um tipo de hospedagem compartilhada projetada especificamente para o WordPress, com otimizações para desempenho e segurança. Dê uma olhada na tabela a seguir para conferir informações mais detalhadas sobre os planos de hospedagem WordPress da Hostinger — desde o preço até os recursos de servidor alocados para o plano:

	WordPress Single	WordPress Starter	WordPress Business	WordPress Pro
<b>Preço</b>	<b>R\$ 6.99/mês</b>	<b>R\$ 9.99/mês</b>	<b>R\$ 13.99/mês</b>	<b>R\$ 39.99/mês</b>
<b>Número de sites</b>	1	100	100	300
<b>Armazenamento em SSD</b>	30 GB	100 GB	200 GB	200 GB
<b>Largura de banda</b>	100 GB	Ilimitada	Ilimitada	Ilimitada
<b>Tráfego mensal</b>	~10.000	~25.000	~100.000	~300.000
<b>Domínio grátis</b>	Não	Sim	Sim	Sim
<b>SSL grátis</b>	Sim	Sim	Sim	Sim
<b>Contas de email</b>	1	100	100	100
<b>Bancos de dados MySQL</b>	2	Ilimitados	Ilimitados	Ilimitados
<b>Backups diários</b>	Não	Não	Sim	Sim
<b>Subdomínios</b>	2	100	100	300

Para ter o melhor desempenho e os melhores tempos de carregamento com a sua hospedagem WordPress, leve em conta os seus objetivos, a quantidade de conteúdo que você quer fazer upload e quanto tráfego você estima que terá.

É natural querer economizar dinheiro sempre que possível, mas a hospedagem deve ser uma prioridade no seu orçamento. É um investimento fundamental que vai determinar quão bom será o desempenho do seu site.

Idealmente, o plano que você escolher terá recursos suficientes para seu uso diário e ainda terá uma margem para upgrades e para dias em que você pode ter picos inesperados de tráfego.

Deste modo, você não terá que se preocupar em de repente ficar sem armazenamento, ou com o seu site caindo quando tiver muito tráfego. Além disso, se você usa a hospedagem compartilhada, ultrapassar os limites dos recursos alocados para o seu plano pode impactar negativamente outros sites no mesmo servidor.

## **Escolha o Servidor Localizado Mais Próximo do seu Público-Alvo**

A localização do seu servidor determina o quão rápido ele pode transferir e receber solicitações para carregar o seu site. Muitos avanços foram feitos para melhorar a velocidade da internet nos últimos anos, mas a latência ainda é um problema.

A latência é o tempo que leva para os servidores da internet receberem e processarem uma solicitação por uma página da web. Um fator que influencia isso de maneira significativa é a distância física para o servidor.

As solicitações da web e os dados viajam de servidor a servidor até que chegam no navegador do usuário. Se um servidor está distante, a latência será alta e levará muito mais tempo para carregar as páginas no navegador.

Uma grande latência deixa os sites mais lentos e piora a experiência geral dos usuários. Isso é ainda pior para um site que tem muito conteúdo pesado, como imagens, animações e vídeos. A latência se torna um problema mais sério para usuários com conexões à internet lentas ou instáveis.

A Hostinger possui seis data centers localizados ao redor do planeta. Quando você assinar um plano de hospedagem, não se esqueça de escolher o servidor que esteja mais próximo do seu público-alvo. Se você já tem um site e quer mudar a localização do seu servidor, contate o suporte.

Contudo, você pode não saber onde está localizada a maior parte do seu público. Para ver um relatório detalhado do **tráfego do seu site**, incluindo a localização dos seus visitantes, use o **Google Analytics**.

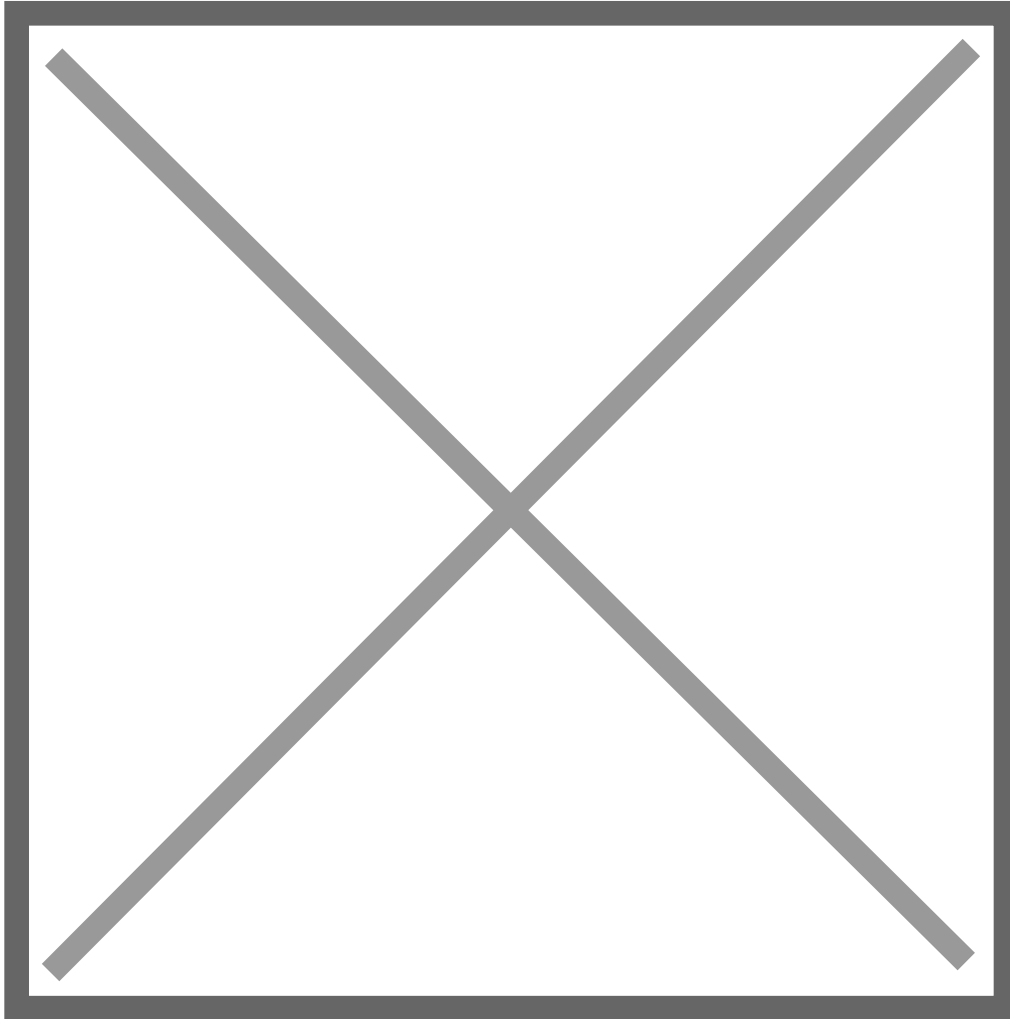
O plugin de Google Analytics mais usado em sites WordPress é o MonsterInsights. **Instalar o MonsterInsights** vai adicionar o Google Analytics ao seu site, e você poderá visualizar as métricas de tráfego diretamente do painel de controle do WordPress.

Se você usa a Hostinger e precisa trocar para um servidor diferente, siga estes passos simples:

1. Acesse seu painel de controle do hPanel, abra a seção de detalhes do Plano e encontre a janela de localização do Servidor. Clique no ícone **Editar**.

A seção de detalhes do Plano no painel hPanel. O ícone Change server é destacado

2. Uma nova janela vai aparecer. No menu suspenso, selecione a localização do servidor para o qual você quer migrar. Note que você só pode iniciar uma transferência de servidor por mês. Clique em **Próximo** para seguir em frente.



Tenha em mente que pode levar até 24 horas para que a migração do servidor seja concluída. Não recomendamos fazer mudanças no seu site WordPress durante o processo, já que elas não ficarão salvas.

Assim que o site tiver sido totalmente transferido para o novo servidor, o endereço de IP associado com o seu plano de hospedagem será renovado.

# Use uma Rede de Fornecimento de Conteúdo

Uma CDN é uma rede de servidores geograficamente distribuídos que trabalham em conjunto para acelerar o processo de entrega de sites aos navegadores dos usuários.

É a solução perfeita para um site WordPress global com públicos-alvo localizados em diferentes locais. Um CDN reduz a distância entre o usuário e o servidor, reduzindo significativamente a latência e aumentando a velocidade da página.

 infográfico explica como funciona uma cdn

Um CDN funciona ao salvar cópias do conteúdo estático de um site no cache. Quando um usuário visita o site, o servidor mais próximo a ele carregará o conteúdo, reduzindo o número de solicitações HTTP externas.

O WordPress funciona bem com várias opções de CDN. Alguns provedores de hospedagem oferecem pacotes de CDN com seus planos.

## Dica

Hostinger oferece sua própria solução CDN: a Hostinger CDN. Ela está disponível para clientes de planos de Hospedagem de Sites Business ou superiores.

## Entregue o Conteúdo Através do HTTP/2

O Protocolo de Transferência de Hipertexto (HTTP) é a base da comunicação de dados na internet. É através dele que clientes e servidores trocam informações.

Por exemplo, quando alguém quer visitar **seudominio.com**, o seu navegador envia uma solicitação HTTP para um servidor remoto que irá processá-la e carregar o conteúdo do site.

A primeira iteração do HTTP, o HTTP/1, é muito efetiva. Contudo, ela é consideravelmente mais lenta em comparação com a versão mais nova, o **HTTP/2**.

Essas são algumas das vantagens do HTTP/2 sobre o HTTP/1:

- **Comunicação eficiente.** O HTTP/2 usa um protocolo binário projetado para a comunicação entre máquinas. Em comparação com o protocolo baseado em texto usado no HTTP/1, os dados binários são significativamente mais simples de se processar.
- **Priorização ponderada.** Com o HTTP/2, o desenvolvedor do site pode decidir qual conteúdo carregar primeiro. Essa funcionalidade permite que os navegadores carreguem antes conteúdos mais leves como texto, dando a impressão de que o site está carregando



mais rápido.

- **Multiplexing.** Com o HTTP/1, o navegador pode apenas carregar um elemento de cada vez. O HTTP/2 melhora isso ao permitir que os navegadores carreguem diferentes tipos de conteúdo numa única conexão.  
processo de multiplexing com http/2
- **Push do servidor.** O HTTP/2 permite que os servidores enviem conteúdo antes do navegador solicitá-lo.
- **Compressão avançada.** A HPACK, que é a compressão que o HTTP/2 usa, pode eliminar a redundância na troca de pacotes HTTP. Essa compressão faz com que as mensagens HTTP sejam menores, o que reduz o tempo de carregamento

Felizmente, a maioria dos provedores de hospedagem WordPress já usam o HTTP/2 por padrão. Não existe necessidade de começar a migração manualmente porque o LiteSpeed, que é o software de servidor usado pela Hostinger, suporta HTTP/2 de modo nativo.

Os provedores de hospedagem WordPress que usam servidores Nginx também suportam HTTP/2 de maneira nativa, enquanto servidores Apache adicionaram suporte para a tecnologia em 2015. Se o seu servidor usa Apache, contate-os para confirmar se o suporte a HTTP/2 está disponível.

Contudo, se você não sabe se o seu site do WordPress usa HTTP/2, utilize a **Ferramenta de Teste HTTP/2** e o **HTTP2.Pro** para verificar. Se o seu site ainda estiver usando HTTP/1, confira se o seu provedor de hospedagem oferece HTTP/2 e entre em contato com eles para iniciar a migração.

Além disso, tenha em mente que instalar um **certificado SSL da Let's Encrypt** é altamente recomendado para o HTTP/2, já que quase todos os navegadores **apenas suportam HTTP/2** através de uma conexão criptografada.

## Dica

Caso você não tenha um SSL instalado no seu site, seja de qualquer tipo, garanta o seu próprio **SSL gratuito** aqui mesmo na Hostinger.

## Escolha um Tema WordPress Otimizado

O tema que você usa no WordPress pode influenciar a velocidade do seu site. É essencial escolher um tema bem otimizado para acelerar as suas páginas.

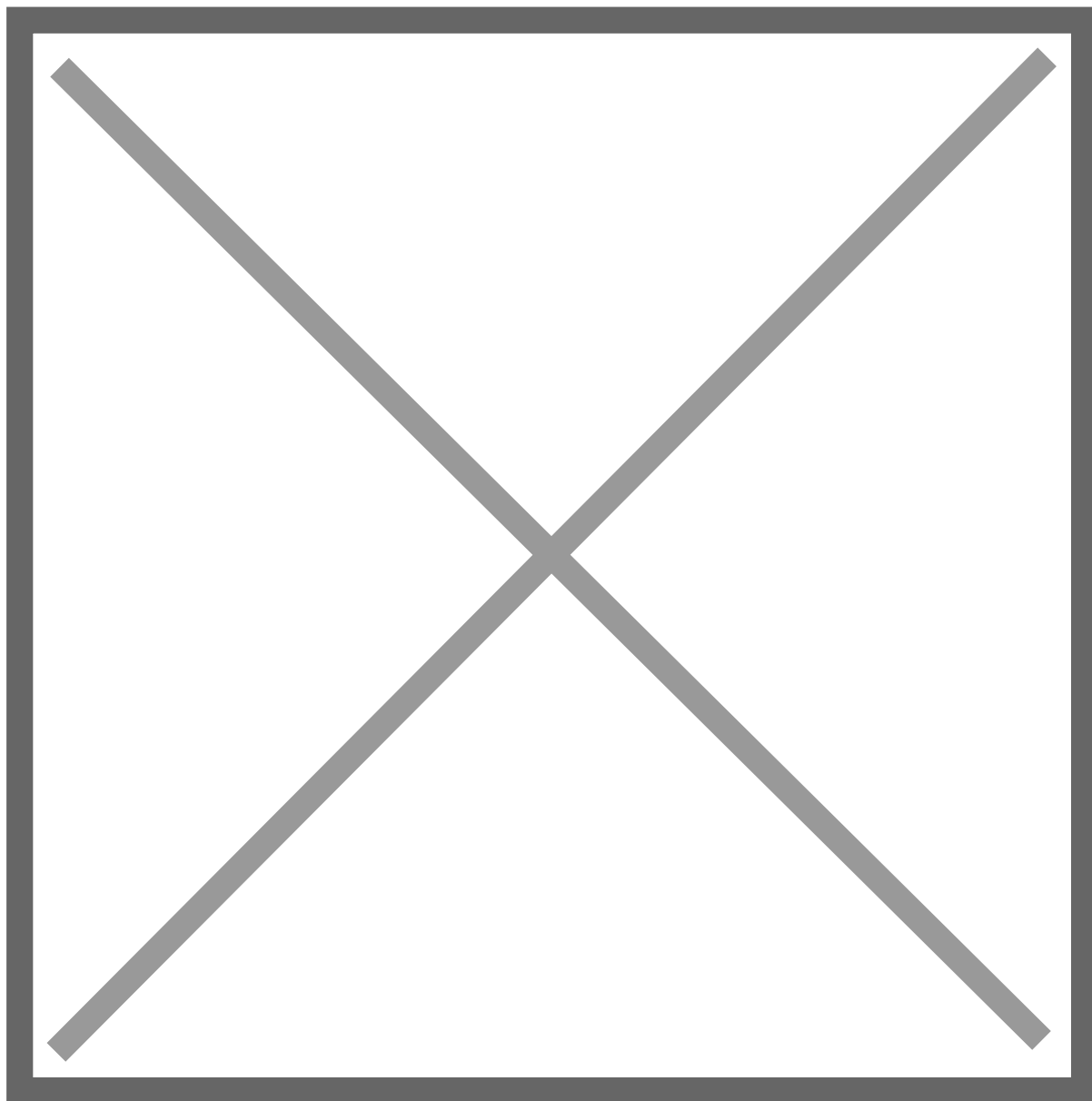
Fique de olho nos seguintes critérios para escolher um **tema WordPress rápido**:

- **Código otimizado.** Para evitar quaisquer erros de desempenho ou segurança, garanta que o código do tema cumpre os padrões do World Wide Web Consortium (W3C) e do WordPress. Use ferramentas como o **HTML Markup Validation Service** e o plugin gratuito **Theme Check** para validar rapidamente o código de um tema.

- **Leveza.** Um tema leve traz apenas os recursos necessários. O seu código deve ser organizado, e os tamanhos dos arquivos JavaScript e CSS idealmente ficam abaixo de **100 KB**. Os temas padrões do WordPress geralmente se encaixam nessa categoria.
- **Compatibilidade com navegadores.** Infelizmente, nem todos os temas do WordPress têm bom desempenho em diferentes navegadores. Escolha um tema compatível com múltiplos navegadores para oferecer a melhor experiência para qualquer usuário.
- **Responsividade.** Quando um tema é responsivo, ele pode se adaptar às telas de dispositivos móveis sem ficar lento. Como **mais da metade de todo o tráfego da internet** se origina de dispositivos móveis, a responsividade é importante para garantir acessibilidade e atrair visitantes.
- **Pronto para AMP.** A tecnologia de **Páginas Móveis Aceleradas** (AMP) permite que os sites do WordPress carreguem mais rapidamente em dispositivos móveis. Escolher um tema pronto para AMP pode deixar seu site significativamente mais rápido.

Antes de instalar um tema, certifique-se de experimentar o site de demonstração dele. Se o tema for gratuito, instale-o e ative-o no seu site, como você faria com qualquer conteúdo adicional. Então, teste sua velocidade usando uma ferramenta como o **PageSpeed Insights**.

Para fazer isso, vá até o site do **PageSpeed Insights**, insira o **domínio personalizado** do seu site e clique em **Analyze**.



Além de pontuar os sites com base em seu desempenho, o PageSpeed Insights também oferece dicas pontuais sobre como melhorar a otimização de velocidade das suas páginas.

O **Pingdom** e o **GTmetrix** são outras **ferramentas de velocidade de site** populares que você pode usar para conferir o desempenho das suas páginas do WordPress com o tema aplicado.

Se o site estiver lento quando for testado, isso significa que o tema que você está usando é mal otimizado. Tente outros temas do WordPress até você atingir a velocidade desejada.

## Dica

Ainda não tem um domínio? A Hostinger tem [registro de domínio internacional](#) e também nacional. Você navega por uma variedade de extensões e escolhe aquela que mais combina com seu negócio. E, se estiver precisando de opções de nomes incríveis, use nosso **gerador de nomes**. A ferramenta usa IA para bolar várias opções de nomes de empresas, sites ou blogs, para vários

nichos diferentes.

## Otimize as Imagens

Imagens são vitais para a estética de um site. Ela são uma das melhores maneiras de atrair atenção, mostrar um produto ou evocar uma emoção em particular.

Contudo, imagens muito pesadas podem aumentar de maneira significativa o tamanho de uma página, fazendo com que o tempo de download fique mais lento. De acordo com o [HTTP Archive](#), em agosto de 2021 as imagens representavam cerca de 45% do tamanho médio total de uma página. Isso é muito mais do que CSS, fontes e arquivos HTML combinados.

Aqui estão algumas dicas para [otimizar as imagens](#) e deixar o WordPress mais rápido:

- **Garanta que arquivo tem 500 KB ou menos.** Antes de fazer o upload de qualquer imagem, confira o tamanho do arquivo. Em geral, imagens com mais de 500 KB carregam de maneira consideravelmente mais lenta. Use uma ferramenta tipo o [TinyPNG](#) para comprimir as imagens.
- **Use o formato de arquivo recomendado.** Diferentes formatos de imagens usam diferentes métodos de compressão de arquivos. Como resultado, alguns formatos de arquivos são mais adequados para situações específicas do que outros. Aqui estão alguns dos formatos de imagem mais comuns e qual o seu melhor uso:
  - **JPEG.** O JPEG usa compressão com perdas, o que significa que alguma informação vai ser perdida para se ter um tamanho de arquivo menor. Os JPEGs são melhores para compartilhar fotografias.
  - **PNG.** O PNG usa compressão sem perdas, que retém toda a informação de uma imagem, garantindo a melhor resolução. Os PNGs são recomendados para ilustrações gráficas e desenhos.
  - **WebP.** O WebP está ganhando popularidade por causa da sua habilidade de combinar compressão com e sem perdas. Ao usar o WebP, você pode oferecer uma imagem de alta resolução que é menor que um arquivo JPEG ou PNG.
- **Use um tamanho de exibição adequado.** Evite fazer upload de imagens de alta resolução e depois [redimensioná-las no WordPress](#). Elas até vão aparecer menores, mas as máquinas ainda precisam fazer o download do arquivo inteiro do servidor, o que ocupa uma maior largura de banda. Se você precisa de imagens com **2000 x 1000 px**, é melhor redimensioná-las para o tamanho exato antes de fazer o upload.
- **Utilize Lazy Loading.** Quando o *lazy loading* está habilitado, as imagens da página vão carregar sob demanda. Isso ajuda a otimizar o site porque evita que ele tenha que carregar todas as fotos de uma só vez, já que elas só serão baixadas quando o usuário rolar a tela para baixo.

Nós recomendamos instalar um plugin de otimização de imagem do tipo tudo-em-um, como é o caso do [WP Smush](#). Isso vai garantir que você tenha uma solução rápida, simples e direta. O WP Smush permite que você execute uma compressão e um redimensionamento em massa das

imagens da sua biblioteca de mídia do WordPress. Além disso, o plugin tem o recurso de *lazy loading*.

Adicionalmente, a versão Pro do WP Smush permite que você converta suas imagens para WebP.

## Leitura Sugerida

Leia nosso artigo sobre a **comparação entre imagens com perda e sem perda** para ver qual é o tipo ideal para melhorar a velocidade de carregamento do seu site!

## Atualize Regularmente o Núcleo, os Plugins e os Temas do WordPress

Ao se manter atualizado com as atualizações de software do WordPress, você vai garantir que o seu site sempre tem as últimas melhorias de desempenho e segurança. A segurança e o desempenho estão interconectados — se um site do WordPress está vulnerável, ele pode ficar mais lento ou até inutilizável.

Como uma plataforma de código aberto, o WordPress infelizmente atrai hackers, que estão constantemente procurando por falhas de segurança. Por causa disso, **manter o WordPress atualizado** é de extrema importância.

Por sorte, desde o **WordPress 5.5**, as atualizações automáticas para o núcleo, os plugins e os temas podem ser habilitadas facilmente direto do painel de controle. Dito isso, você ainda tem a opção de atualizá-los manualmente.

No caso do núcleo do WordPress, você pode habilitar atualizações automáticas apenas para upgrades menores, que é o recomendado para a maioria dos usuários. Grandes atualizações podem adicionar mudanças significativas que podem não combinar com o seu site.

Para escolher como o software núcleo do WordPress deve ser atualizado automaticamente, faça login e navegue até **Atualizações**.

- Se os updates automáticos menores já estiverem habilitados, você verá uma mensagem que diz **Este site é automaticamente atualizado apenas com versões de manutenção e segurança do WordPress**.
- Se você quiser habilitar a atualização automática para grandes updates, clique em **Ativar atualizações automáticas para todas as novas versões do WordPress**. Assim que essa opção estiver habilitada, aparecerá a mensagem **Este site é automaticamente atualizado para cada nova versão do WordPress**.

De modo alternativo, se você usa a Hostinger, tem a opção de habilitar as atualizações automáticas de núcleo do WordPress a partir do hPanel. Para fazer isso, vá até **WordPress → Visão Geral** na barra lateral. Na área **Núcleo**, encontre **Atualizações do WordPress** e clique em **Alterar**.

Na nova janela, ative a opção **Ativar atualizações automáticas**. Se quiser, clique em **Mostrar configurações avançadas** e escolha as opções de atualização automática que você preferir. Você pode configurar separadamente as atualizações do WordPress, dos temas e dos plugins, podendo optar por ativar todas as atualizações, apenas atualizações de segurança ou desativar as atualizações de cada um deles.

ligando as atualizações automáticas do wordpress no hpanel

Para atualizar um plugin, faça login no painel de controle e vá até **Plugins → Plugins Instalados**. Encontre o plugin que você quer atualizar e selecione **Atualizar agora**. Se você preferir, também dá para escolher **Ativar atualizações automáticas** na coluna **Atualizações Automáticas**.

atualizando plugins do wordpress pelo hpanel

Ainda assim, por segurança, sempre recomendamos que você leia o registro de mudanças antes de atualizar qualquer plugin. Isso é especialmente vital se você tem um grande site que precisa lidar com muito tráfego.

As atualizações de segurança normalmente são seguras para serem aplicadas imediatamente. Contudo, as atualizações de recursos podem ter bugs que afetam a usabilidade e o desempenho. Uma boa regra para seguir é esperar uma semana antes de aplicar qualquer atualização. Até lá, quaisquer problemas conhecidos já deverão ter sido resolvidos pelos desenvolvedores.

Para atualizar um **tema**, vá até **Temas Instalados** e clique no template que você quer atualizar. Para atualizar o tema manualmente, clique em **Atualizar**.

atualizando temas do wordpress no hpanel

Assim como acontece com os plugins, tenha cuidado na hora de aplicar as atualizações. Também é altamente recomendado conferir as notas da nova versão antes de atualizar os temas.

Outra dica é usar um **tema filho**. Um tema filho herda todos os recursos do tema pai, mas permite que os usuários do WordPress o modifique sem promover alterações ao tema pai. Como resultado, quando você for aplicar atualizações ao *parent theme*, será possível manter as alterações feitas ao *child theme*.

Se for uma grande atualização, pode ser uma boa ideia **configurar um ambiente de teste** para ver se está tudo certo antes de aplicá-la ao seu site. Isso pode ser feito manualmente ou usando um plugin tipo o **WP Staging**.

## Desinstale Plugins Desativados

Os plugins expandem as funcionalidades do WordPress, o que os torna ótimas adições para um site. Contudo, alguns plugins podem demandar muitos recursos, atrapalhando o desempenho. É por isso que você deve evitar ter muitos plugins do WordPress, e deve remover os que não usa

mais.

Desativar um plugin não é suficiente porque ele ainda fica ligado ao seu site. Você tem que deletar totalmente os plugins não usados.

Felizmente, é bem simples de remover a maioria dos plugins. No seu painel de controle, vá até **Plugins**, localize o plugin desativado e selecione **Remover**.

Plugins inativos podem deixar algumas consultas de bancos de dados para trás e configurações do WordPress. Isso requer que você tome alguns passos extras para removê-los completamente. Trata-se de algo comum com plugins de segurança ou de cache.

Antes de **remover um plugin desativado**, confira a documentação oficial dele — que vai explicar como desinstalá-lo do jeito certo. Por exemplo, o Yoast SEO tem um artigo bem abrangente (em inglês) sobre como **desinstalar completamente plugin deles**.

Se o plugin não possui um guia aprofundado de desinstalação, você terá que **remover manualmente** os arquivos de banco de dados e as configurações que ficarem para trás.

Para evitar ter que desinstalar os plugins do WordPress com frequência, tente ser mais seletivo e antes de instalá-los:

- **Sempre pesquise sobre o desenvolvedor.** Idealmente, o desenvolvedor deve ter criado bons plugins no passado e responder ao feedback dos usuários. Ele também deve atualizar o plugin regularmente.
- **Use marketplaces confiáveis.** Se você quer instalar um plugin de alguma fonte alternativa, tenha certeza que ela é confiável, como no caso do **CodeCanyon**. Fontes não confiáveis têm maior chance de incluir plugins de baixa qualidade, que pode até estar contaminados com vírus.
- **Apenas instale os plugins necessários.** Antes de instalar qualquer plugin, pergunte a si mesmo se ele é essencial para o seu site. Quanto menos plugins você tiver, mais leve e rápido será o seu website do WordPress.

Menos plugins resultam em melhor desempenho. Contudo, a qualidade do plugin é igualmente importante. Uma opção de baixa qualidade pode causar problemas de desempenho devastadores. Portanto, é imperativo que você faça uma pesquisa adequada antes de baixar um plugin.

## Minifique o HTML, o CSS e o JavaScript

Se você já abriu um arquivo CSS, HTML ou JavaScript, vai perceber que alguns deles possuem muitos espaços, quebras de linhas, comentários ou caracteres redundantes. Isso pode até torná-los mais legíveis para outros desenvolvedores, mas acaba sendo uma desorganização que faz o site carregar mais devagar.

Minificar arquivos CSS, HTML e JavaScript significa remover caracteres desnecessários para reduzir os tamanhos dos arquivos. Você pode otimizar o WordPress com arquivos menores, já que as transferências de dados entre servidores serão mais rápidas.

Existem duas maneiras principais de minificar um arquivo JavaScript, HTML ou CSS:

- **Use uma ferramenta online.** O site **MinifyCode** tem ferramentas que permitem minificar automaticamente o conteúdo de arquivos JavaScript, CSS e HTML. Tudo o que você precisa fazer é substituir os arquivos antigos com versões minificadas.
- **Instale um plugin de minificação.** Diversos plugins permitem que você minifique arquivos HTML, JavaScript e CSS — um deles é o **Fast Velocity Minify**. Nas configurações do plugin, simplesmente selecione os tipos de arquivo que deseja minificar.

A minificação de arquivos é uma das maneiras mais fáceis e eficientes de otimizar o desempenho de um site. Ela frequentemente reduz o tamanho dos arquivos em cerca de 40% a 60%. Plugins de cache como LiteSpeed, W3 Total Cache e Autoptimize também oferecem funcionalidades de minificação.

## Use a Versão PHP Mais Recente

O PHP é uma linguagem de script que permite que desenvolvedores criem sites dinâmicos e interativos. De acordo com o W3Techs, o PHP é usado em **78,4% de todos os sites** com uma linguagem de script de servidor identificável. O CMS WordPress e todos os sites construídos nele são escritos em PHP.

Como qualquer outro software, o PHP tem sido constantemente atualizado desde o seu primeiro lançamento. Cada nova versão do PHP introduz melhorias de performance e segurança. A versão mais recente do PHP é a 8.0, que também é suportada pelos planos de hospedagem da Hostinger.

Segundo o benchmark da Phoronix, o PHP 8.0 **tem um desempenho 10% superior** ao seu predecessor, o PHP 7.4. Ele vem com muitos novos recursos e otimizações, melhorando a eficiência da linguagem.

Um recurso-chave do PHP 8.0 é a compilação just-in-time (JIT). Em termos simples, a JIT permite que o PHP 8.0 compile um programa em código de máquina imediatamente antes da execução. Sem a compilação JIT habilitada, o desempenho do PHP 8.0 no WordPress é **ainda mais rápido**.

Atualmente, o WordPress **suporta o PHP 7.3** ou superior. Infelizmente, **grande parte dos sites feitos em WordPress — 32,38%** para ser exato — ainda estão usando uma versão do PHP que não é mais suportada pelo CMS. Além disso, somente **1,54%** dos sites do WordPress usam a versão mais recente do PHP.

Se você ainda está usando uma versão antiga do PHP, irá ficar de fora de recursos novos, melhorias de desempenho e atualizações de segurança. Além de melhorar a velocidade do seu



site, atualizar o PHP também vai melhorar a segurança em geral.

A maioria dos provedores de hospedagem WordPress permite que você altere a sua versão do PHP diretamente do painel de controle da hospedagem. Para mudar a versão do PHP na Hostinger, siga estes passos:

1. Faça login no hPanel e navegue para **Avançado → Configuração PHP**.

O botão de configuração PHP no hPanel

2. Na aba **versão PHP**, selecione a sua versão preferida e clique em **Atualizar**.

A seção de Configuração PHP no hPanel. A seção de Opções PHP é mostrada

Antes de atualizar a versão do PHP, garanta que o código do seu site do WordPress é compatível com a versão que você está instalando. É recomendado que você primeiro tente fazer o upgrade da versão PHP num ambiente de testes. Essa é a melhor maneira de diminuir o risco do seu site ter problemas técnicos.

## Otimizar Banco de Dados

O banco de dados do seu site armazena todos os tipos de dados e informações que chegam até ele. Isso inclui conteúdos, logins, senhas, arquivos baixos e enviados a servidores, comentários e spams. É por isso que você precisa limpá-lo de vez que quando.

Da mesma forma como fizemos antes com a desinstalação de plugins e temas desnecessários, apagar estes tipos de dados vai diminuir o peso exigido sobre o banco de dados e melhorar o desempenho geral do seu site.

Para fazer isso, você pode instalar um plugin de otimização de banco de dados, como o **WP-Optimize** ou o **Advanced Database Cleaner**. Ambos vão automaticamente remover dados desnecessários do seu site, além de compactar tabelas e imagens numa hora programada.

## Recomendações de Leitura

Se você está em busca de mais maneiras para melhorar o desempenho do WordPress, estes artigos vão de ajudar a encontrar o caminho certo:

- **30 dicas de SEO para seu site WordPress**
- **Como deixar seu site responsivo com Google AMP**
- **Por que você precisa de Schema e como usar no WordPress**

## Crie Seu Site WordPress Para o Futuro

Sites que são capazes de se adaptar a melhorias tecnológicas têm maior probabilidade de sobreviver na próxima década do mundo web. Da mesma forma, desenvolvedores que querem conquistar o sucesso devem estar preparados para mudanças.

Fique sempre atento às novas atualizações, plugins e tendências para otimizar WordPress, para que você possa estar sempre na frente. O desempenho do WordPress pode ser complicado às vezes, mas ao implementar as dicas acima, você já terá uma vantagem competitiva em relação aos concorrentes.