Engineer

WordPress

gestor de contenidos web de código abierto

Qué es

WordPress es un gestor de contenidos de código abierto: software PHP que almacena las entradas, páginas, registros de medios, usuarios y ajustes de un sitio en una base de datos MySQL o MariaDB y compone las páginas a partir de ellos cuando se solicitan. Los archivos multimedia subidos suelen estar fuera de la base de datos. Su nombre se abrevia a menudo como WP, el mismo prefijo que llevan sus archivos y carpetas (wp-config.php, wp-content, wp-admin). Se distribuye bajo GPLv2 o posterior, por lo que descargar, ejecutar o modificar el software no cuesta nada.

Un sitio WordPress tiene tres capas. El núcleo es el código común que todos descargan y actualizan. El tema controla la presentación y los plugins añaden comportamiento: ahí traza la separación el propio manual de temas del proyecto. Ambos intervienen en el núcleo mediante hooks, puntos con nombre del código WordPress donde se permite ejecutar otro código. Hay dos clases: acciones y filtros. Una acción se ejecuta en un momento concreto, como al guardar una entrada, y no devuelve nada. Un filtro recibe un valor, lo modifica y lo devuelve. Este reduce los extractos automáticos de las entradas de las 55 palabras predeterminadas a 30:

```
add_filter( 'excerpt_length', function ( $length ) {
    return 30;
} );
```

Los temas han cambiado de forma con los años. Los clásicos utilizan el sistema de plantillas PHP que llegó con WordPress 1.5 en 2005 y siguen muy extendidos. Los temas de bloques son el enfoque actual: sus plantillas son archivos HTML

compuestos de bloques que se pueden editar en Site Editor, con colores, fuentes y espaciados en un archivo theme.json. Los temas híbridos son clásicos que han incorporado parte de las funciones de bloques.

El contenido también sale del sitio como JSON mediante la API REST, sobre la que está construido el propio editor de bloques. Una petición a la dirección /wp-json/ del sitio devuelve un mapa de todas las rutas que ofrece. El contenido público puede leerse sin iniciar sesión y todo lo privado requiere autenticación. Cualquier programa puede obtener así las tres entradas más recientes:

```
curl -s 'https://example.com/wp-json/wp/v2/posts?per_page=3'
```

Para ejecutarlo, el proyecto recomienda PHP 8.3 o superior, MySQL 8.0 o MariaDB 10.11 o superiores y HTTPS, preferiblemente sobre Apache o Nginx. Joomla y Drupal son gestores de contenidos del mismo tipo. Las otras dos vías con las que compararlo son una aplicación a medida sobre un framework como Django y un generador de sitios estáticos.

Utilice el concepto en una decisión

Aplique «¿Qué es WordPress?» a una decisión real. Podemos ayudarle a relacionar el concepto con el sistema que está construyendo o comprando. Un sitio rápido, fácil de encontrar y editable por su propio personal, sin una factura por cada cambio de texto.

Explore nuestros servicios Póngase en contacto →

Compruebe los supuestos detrás del término

Utilice la explicación y los trabajos enlazados para formular preguntas sobre su proyecto. Podemos examinar los supuestos con usted antes de elegir una herramienta. 100 aplicaciones web con HTML/HTML5 — 100 aplicaciones web desarrolladas

100 aplicaciones web con HTML/HTML5 Póngase en contacto →

Por qué importa

La decisión real detrás de WordPress es quién cambia el sitio después del lanzamiento y quién lo mantiene seguro mientras lo hace. WordPress ofrece al personal no técnico un editor completo: pueden publicar una página, cambiar una imagen, añadir un usuario o rehacer un menú sin abrir una solicitud de trabajo. En un sitio que cambia cada semana, la modificación más barata es la que no necesita a un desarrollador.

La licencia es gratuita; el sitio tiene costes. La GPL cubre el software, y wordpress.org sostiene que los plugins y temas son obras derivadas que heredan la misma licencia. El gasto se desplaza a otras partidas: alojamiento que ejecute PHP y una base de datos, horas para elegir, configurar y probar un tema y trabajo continuo para mantener actualizados el núcleo, los plugins, los temas y la versión de PHP. Ese último coste dura toda la vida del sitio, y un presupuesto debería indicar quién lo asume.

El mantenimiento se deriva de que WordPress es una aplicación activa. Para generar una página que no está en caché se ejecuta PHP contra una base de datos; una caché de página completa puede servir una página guardada. El código de los plugins se ejecuta con el mismo acceso que el núcleo, y la pantalla de acceso está expuesta a internet salvo que alguien ponga una segunda cerradura delante. Las actualizaciones del núcleo en segundo plano llegaron en la versión 3.7. Las instalaciones nuevas a partir de la 5.6 activan de forma predeterminada tanto actualizaciones menores como mayores del núcleo, según la configuración y las condiciones para poder actualizar. Las instalaciones anteriores conservan sus ajustes previos salvo que se cambien. PHP sigue su propio calendario de actualizaciones.

Una solución a medida encaja con datos y procesos que no se parecen en nada a páginas y entradas, y sus procesos de edición deben construirse y mantenerse. Una generación estática saca a la aplicación PHP y la base de datos de la ruta que atiende las peticiones, aunque el servidor y las herramientas de generación siguen necesitando mantenimiento. Sus editores suelen trabajar mediante archivos y un paso de generación. WordPress encaja en el amplio espacio intermedio: un sitio bastante convencional que sus propietarios gestionarán por sí mismos y cuyas pantallas de edición compensan el mantenimiento que traen. La dependencia de la plataforma es moderada: el código es GPL, el contenido reside en una base de datos SQL corriente y la API REST entrega entradas, páginas y taxonomías como JSON.

Cómo se aplica en la práctica

En el trabajo de esta empresa, WordPress es una de varias respuestas, elegida deliberadamente. La descripción de cien aplicaciones web abarca 2003–2014 y expone la regla de elección. Cuando un cliente necesitaba algo a medida, se construía con HTML/HTML5, CSS/SCSS y Django. Cuando necesitaba un sitio más convencional que también pudiera gestionar por sí mismo, WordPress o Joomla permitían llegar antes y tenían más sentido.

Lo que valora esa descripción es el criterio, no la herramienta. Parte del trabajo consistía en distinguir ambos casos: disuadir a un cliente de una solución a medida que no necesitaba y convencerlo de elegirla cuando la necesidad era real.

Ese criterio, expresado como preguntas que quien contrata puede responder antes de solicitar un presupuesto, sería el siguiente:

- ¿Editarían el sitio cada semana personas sin ayuda de un desarrollador, o cambiaría solo unas pocas veces al año?
- ¿Encaja el contenido en páginas, entradas, categorías y medios, o necesita un modelo de datos y unas reglas propios?
- ¿Quién mantendrá actualizados el núcleo, los plugins y PHP mientras el sitio siga activo?
- ¿Se apoyará el sitio en unos pocos plugins bien mantenidos o en docenas de ellos unidos para imitar una aplicación?

Las ediciones semanales, un contenido que encaja, una persona designada como responsable del mantenimiento y una lista corta de plugins apuntan a WordPress o a un CMS similar. Los datos poco habituales o la falta de alguien que lo mantenga actualizado apuntan a una solución a medida o estática.

Dónde surgen los problemas

Una actualización rutinaria rompe el sitio

Alguien actualiza un plugin y una página empieza a mostrar un error o el panel de administración no carga. El sitio había acumulado plugins durante años, algunos abandonados por sus autores, cada uno con PHP de terceros con acceso completo, y dos entraron en conflicto. Mantenga corta la lista y elimine todo plugin que el sitio haya dejado de utilizar, como pide el manual de refuerzo de seguridad. Haga una copia de seguridad antes de cada actualización, pruebe primero las actualizaciones mayores en una copia y prefiera plugins cuyos autores sigan publicando versiones.

Los cambios propios desaparecen tras una actualización o un rediseño

Un ajuste que funcionó durante meses desaparece el día en que se actualiza el tema, o una función de la que depende el negocio falta cuando cambia el aspecto del sitio. En el primer caso, alguien editó los archivos de un tema ajeno y la actualización los sustituyó; el manual de temas advierte que una función añadida

directamente al functions.php del tema desaparece con su siguiente actualización. En el segundo, un comportamiento esencial para el sitio estaba dentro del tema y se fue con él. Ponga los cambios de un tema ajeno en un tema hijo y las funciones en un pequeño plugin propio que se conecte al núcleo mediante hooks, para que el tema solo se encargue de la presentación.

Una contraseña de administrador se convierte en código en el servidor

Quien roba la contraseña de un administrador puede pasar de editar entradas a ejecutar su propio PHP, porque el panel incluye un editor de archivos de temas y plugins. Una línea en wp-config.php desactiva ese editor y tiene el mismo efecto que retirar a todos los usuarios las capacidades edit_themes, edit_plugins y edit_files:

```
define( 'DISALLOW_FILE_EDIT', true );
```

Cierra una vía entre varias. El manual dice expresamente que no detendrá a un atacante capaz de subir archivos maliciosos. Combine esta medida con contraseñas robustas, autenticación en dos pasos para todos los administradores y el menor número de cuentas de administrador que permita el trabajo.

El sitio se queda atrás respecto a la versión actual

El panel muestra durante meses un aviso de nueva versión mientras el sitio sigue en una versión mayor, o en una versión de PHP, muy atrasada respecto a la actual. Un sitio configurado solo para actualizaciones menores del núcleo puede seguir aplicándose parches mientras la versión mayor y PHP se quedan atrás, hasta que un plugin del que depende exige algo más reciente. Las instalaciones nuevas desde la 5.6 también activan las actualizaciones mayores del núcleo de forma predeterminada, pero la configuración puede cambiarlo. Programe las actualizaciones mayores en el calendario, compruebe al mismo tiempo la versión de PHP en la página de requisitos del proyecto y pruebe ambas en una copia antes de actualizar el sitio en producción.

Lecturas adicionales

- WordPress Documentation — manual del proyecto para gestionar un sitio, desde publicaciones y medios hasta instalación y mantenimiento.

- Requirements — versiones de PHP, MySQL y MariaDB que recomienda actualmente el proyecto.
- Upgrading WordPress — actualizaciones en segundo plano, con un clic y manuales, con la copia de seguridad que debe hacerse primero.
- Hardening WordPress — permisos de archivos y de base de datos, desactivación del editor de archivos y mantenimiento de la lista de plugins.
- Plugin Handbook: Hooks — cómo las acciones y los filtros permiten modificar WordPress sin tocar el núcleo.
- REST API Handbook — rutas, puntos de acceso y autenticación de la API sobre la que se construye el editor de bloques.

Glosario

Elija según sus restricciones reales

Describa el flujo de trabajo, las interfaces existentes y los usuarios que necesitan el cambio. Podemos definir una entrega comprobable y un traspaso que permita a su equipo mantener lo entregado.

Vea cuánto cuesta Póngase en contacto →

Haga práctica la terminología

Pregunte cómo se aplica «¿Qué es WordPress?» a su organización. Basta un ejemplo concreto de su sistema actual para iniciar la conversación.

Explore nuestros servicios

welcome@engineer.company +45 50 10 61 84 Copenhague, Dinamarca

Llévese esta página

Tarjeta de contacto

Creado con esmero y cariño por D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Derechos de autor © 2025 – actualidad · Engineer ApS · Todos los derechos reservados.

Impreso desde https://engineer.company/es/glossary/wordpress/