Engineer

Generador de sitios estáticos

Programa que genera todas las páginas de antemano

Qué es

Un generador de sitios estáticos es una herramienta de construcción que toma una carpeta de archivos fuente, normalmente páginas escritas en Markdown y un conjunto de plantillas, y los convierte en un sitio web terminado compuesto únicamente por archivos: HTML, hojas de estilo, scripts e imágenes que un servidor web entrega tal como están. El término suele abreviarse como SSG, y «estático» describe cómo se entrega el resultado, los mismos bytes a cualquiera que solicite una dirección, no cómo se comporta una página una vez que llega al navegador.

Se contrapone al enfoque dinámico de un sistema de gestión de contenidos o una aplicación web, donde un programa del servidor compone cada página cuando se solicita, normalmente consultando una base de datos y rellenando plantillas. Un generador ejecuta el mismo tipo de plantillas sobre el mismo tipo de contenido, pero una vez por cambio en lugar de una vez por visita, y deja el resultado en una carpeta que un servidor web corriente, un bucket de almacenamiento o una red de distribución de contenidos pueden servir sin saber cómo se hicieron las páginas.

Hugo es uno de esos generadores. Su documentación lo presenta como «un generador de sitios estáticos escrito en Go, optimizado para la velocidad y diseñado para la flexibilidad» que «genera un sitio completo en segundos, a menudo en menos». El ciclo habitual consiste en escribir mientras su servidor web integrado está en ejecución, lo que reconstruye el sitio con cada cambio y recarga las páginas abiertas en el navegador, y después generar el sitio y desplegarlo en un servidor:

```
hugo server     # build, serve, rebuild and reload on every save
hugo --minify   # write the finished site into public/
```

Utilice el concepto en una decisión

Aplique «¿Qué es un generador de sitios estáticos?» a una decisión real. Podemos ayudarle a relacionar el concepto con el sistema que está construyendo o comprando. Interfaces que cargan rápido, funcionan en el teléfono y se comportan igual en los navegadores que utilizan sus clientes.

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. Dirección del desarrollo integral de mapas GIS — cartografía integral, de la base de datos al navegador

Dirección del desarrollo integral de mapas GIS Póngase en contacto →

Por qué importa

La elección determina cuándo se hace el trabajo. Un generador lo traslada del momento de la solicitud al de la construcción, y casi todo lo demás se deriva de ello.

Enviar un archivo es una de las operaciones menos costosas para un servidor web, por lo que un sitio generado responde con rapidez y puede funcionar en una máquina muy pequeña. Como cualquier visitante de una dirección recibe los mismos bytes, el almacenamiento en caché es sencillo. Sin código de aplicación ni base de datos en ejecución detrás de las páginas, hay mucho menos que proteger y actualizar. El resultado es una carpeta, así que cambiar de servidor consiste en copiar archivos, y el código fuente está bajo control de versiones, donde se puede revisar y rastrear cada cambio de cada página.

Los costes proceden del mismo lugar. Cada cambio, incluida la corrección de una errata, exige una nueva construcción y un despliegue, por lo que la velocidad de construcción importa más a medida que crece el número de páginas. Todo lo que dependa de quién pregunta, como las búsquedas, los comentarios, el inicio de sesión o un formulario que almacene lo que recibe, necesita código en el navegador, un servicio independiente o una forma de resolverse durante la construcción, porque no hay ningún programa en el servidor que lo haga. El tiempo se detiene entre construcciones: Hugo omite por defecto una página cuya fecha esté en el futuro, de modo que una página programada para el lunes aparece con la primera construcción que se haga ese día o después. Incluso las redirecciones cambian de forma. Al no haber un programa que envíe una redirección HTTP, Hugo crea por defecto un pequeño archivo HTML en la dirección antigua de una página trasladada que hace que el navegador continúe hacia la nueva. Su documentación describe este enfoque como compatible con cualquier proveedor de alojamiento.

Cómo se aplica en la práctica

Este sitio se construye con uno. La página de créditos identifica a Hugo como generador de sitios estáticos y le atribuye las plantillas, las taxonomías, todos los idiomas y los canales de sindicación. Sobre lo que llega al navegador, explica que el HTML se escribe por completo antes de que nadie lo solicite: cada página es un archivo en disco, en lugar de una plantilla compuesta para cada solicitud. Incluso el aspecto se decide durante la construcción, porque el fondo del tema claro se sortea mientras se construye la página y el sitio cambia de apariencia con cada publicación. Las comprobaciones de calidad se ejecutan localmente, en el ordenador donde se edita el sitio, sin ningún servicio de construcción alojado en toda la cadena; cualquier diagnóstico se considera un error y, en octubre de 2026, el sitio generaba 2.268 páginas. Se sirven desde un servidor de un solo núcleo y 512 MB de memoria, donde el servidor web y los otros tres servicios que responden por el sitio leen la misma carpeta de archivos, por lo que un despliegue actualiza los cuatro.

La entrada del portafolio sobre la construcción de este sitio con Hugo documenta el estado de la implementación en agosto de 2026 y explica los motivos. Ofrecer tres idiomas, evitar un servicio de aplicación facturado por separado y no poner a los visitantes en manos de terceros descartaba casi todos los sistemas de gestión de contenidos, porque hay que proteger y actualizar un backend en ejecución, pagar por él y explicarlo en la página de privacidad. Entonces el sitio tenía 108 archivos de plantillas, 53 de ellos parciales, publicaba cada página en cuatro representaciones a partir de un árbol de contenido y construía ese árbol siete veces más para subdominios especializados, indicando al generador una carpeta de contenido distinta. La entrada documenta la contrapartida de añadir funciones durante la construcción en lugar de hacerlo en el navegador: salvo la excepción documentada y acotada de JavaScript para un filtro del portafolio, las funciones interactivas tenían que resolverse durante la construcción u omitirse, por lo que se calculó el coste de un índice de búsqueda y se dejó aparcado en lugar de publicarlo.

Dónde surgen los problemas

Una página eliminada sigue en línea. Una página retirada de los archivos fuente sigue sirviéndose en su dirección antigua y los buscadores continúan encontrándola. Hugo no vacía su carpeta de salida antes de construir: sobrescribe los archivos existentes, pero nunca los elimina, por lo que `public/` puede conservar páginas de construcciones anteriores, incluidos borradores y páginas caducadas, y cualquier proceso que copie esa carpeta a un servidor también las

copia. Su caché `resources/` presenta un problema parecido: las reglas de diseño de este sitio advierten de que puede entregar un paquete CSS obsoleto cuando solo ha cambiado el CSS fuente. Genere el sitio en una carpeta limpia, con la opción `--cleanDestinationDir` de Hugo o con un paso de borrado propio; la construcción de este sitio elimina tanto `public/` como `resources/` antes de empezar.

Una página terminada nunca aparece. Se escribe una entrada, se registra en un commit y se despliega, la construcción informa de que ha terminado correctamente y el sitio no contiene esa página. Por defecto, Hugo omite el contenido marcado como borrador, aquel cuya fecha o fecha de publicación está en el futuro y aquel cuya fecha de caducidad ya ha pasado; nada de ello se considera un error. Basta con introducir una fecha adelantada un día. Compruebe primero esos cuatro campos, reserve `--buildDrafts`, `--buildFuture` y `--buildExpired` para las vistas previas y programe una construcción posterior a la fecha de una página programada.

Todas las páginas afirman haber cambiado a la vez. El mapa del sitio asigna a casi todas las direcciones la misma fecha de última modificación, lo que indica al rastreador que ha cambiado todo el sitio sin decirle qué páginas han cambiado. Un generador tiene que obtener de algún lugar la fecha de modificación de cada página, y la marca temporal del propio archivo es una fuente incorrecta cuando un paso reescribe archivos cuyo contenido permanece igual. La configuración de este sitio documenta que su paso de sincronización reescribe los 189 archivos de contenido en cada ejecución, por lo que 172 de las 176 direcciones del mapa compartían una marca temporal. Obtenga la fecha del control de versiones: Hugo lee la fecha de la última revisión Git de un archivo cuando está activado `enableGitInfo`, y una nueva sincronización que produce bytes idénticos no crea ningún commit, de modo que una sincronización posterior a un mes de cambios solo modificó 8 de 186 archivos.

Se publican archivos que nadie pretendía publicar. Cualquiera que adivine su nombre puede descargar un archivo que enumera otros archivos. Un generador publica todo lo que encuentre en las carpetas que lee, salvo lo que retire una regla de exclusión, y una regla que no abarque todas las carpetas deja expuesto el resto. La configuración de este sitio documenta que su lista general de exclusiones no llegaba a un montaje de módulo: la construcción publicó en todos los servidores los seis archivos `.DS_Store` que Finder de macOS había dejado en icons/, images/ y webmanifest/, y cada uno enumeraba los nombres de los elementos de su carpeta, por lo que el de images/ revelaba el nombre de una carpeta que el sitio no publica. Un filtro al que le faltaba un espacio después de

`!` llegó a excluir todas las imágenes de un montaje. Aplique exclusiones en cada montaje que publique contenido y compruebe después el resultado generado en busca de archivos que no deban estar; dos comprobaciones de este sitio rechazan una construcción que sirva alguno.

Lecturas adicionales

- Hugo: Introduction — qué es Hugo, para qué se utiliza y cómo funciona su ciclo de escritura, construcción y despliegue.
- Hugo: Basic usage — cómo construir y servir un sitio, por qué `public/` conserva archivos antiguos y qué contenido omite una construcción.
- Hugo: Configure front matter — cómo elige Hugo las fechas de una página, incluidos los identificadores `:git` y `:fileModTime`.
- Hugo: Configure modules — los montajes, el filtro `files` que acepta cada uno y la regla por la que definir un montaje elimina el predeterminado.
- Hugo: URL management — los alias y los archivos de redirección que un alojamiento estático sirve en lugar de una redirección del servidor.

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 un generador de sitios estáticos?» 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/ssg/