Cómo instalar esta herramienta

Añade herramientas para tus visitantes, con pasos claros y archivos que puedes alojar en tu propio sitio.

Guía 1.0.1 · 31 de octubre de 2026
Paquetes comerciales de esquema 3

Tu ruta de instalación

Prepara el sitio. Sube la pareja JS/WASM. Añade un bloque. Compruébalo antes de publicarlo para todos.

1 Antes de cambiar tu sitio web
2 Sube los archivos. Añade un script.
3 Compruébalo en una página real
4 Ajustes opcionales y sencillos
5 Alojamiento y seguridad
6 Tu sitio sigue necesitando revisión
7 Actualizar, revertir o retirar
8 Solución de problemas
9 Más comprobaciones y referencias

Para propietarios y administradores web. Las primeras secciones cubren la instalación; las siguientes ayudan a desarrolladores y responsables de alojamiento. Puedes leer el HTML sin conexión o usar este PDF impreso.

Antes de cambiar tu sitio web

Necesitas el ZIP comprado, acceso a los archivos del sitio y un lugar para añadir un módulo JavaScript a su plantilla compartida.

1. **Conserva una copia original.** Guarda el ZIP, su licencia y la plantilla actual. Haz una copia de seguridad con tu herramienta habitual y confirma cómo restaurarla.
2. **Comprueba el dominio.** Abre `build-configuration.json` en un editor de texto. Comprueba `schemaVersion: 3`, `licenseProfile: "commercial"` y que `licensedBaseDomain` corresponde al dominio comprado.
3. **Elige una página de prueba segura.** Usa una página restringida o un subdominio de pruebas bajo esa base autorizada. No introduzcas datos reales de clientes en los campos de prueba.
4. **Revisa las capacidades del alojamiento.** Debe permitir servir ambos archivos y ejecutar un módulo ES. Si elimina scripts o bloquea archivos `.wasm`, consulta a su administrador antes de instalar.

Contenido del ZIP comprado

`accesskit_widget_wasm.js`
El módulo JavaScript que carga el widget.

`accesskit_widget_wasm_bg.wasm`
El widget compilado correspondiente. Conserva ambos juntos.

`build-configuration.json`
Dominio, perfil, política e identidad del código de esta compilación.

`checksums.json`
Huellas SHA-256 para comprobar la integridad del paquete.

`INSTALL.txt`
Notas breves de instalación.

`LICENSE.txt` **y** `THIRD-PARTY-NOTICES.txt`
Consérvalos con el paquete y respeta los términos incluidos.

> **El dominio forma parte de la compilación**
>
> En una licencia ilustrativa para `example.com`, el dominio raíz, `www.example.com` y `staging.example.com` comparten la base autorizada. No se incluyen `example.net`, `localhost`, direcciones IP ni un dominio de vista previa del proveedor. Las bases de clientes en servicios compartidos pueden ser distintas. Usa la base exacta del paquete; no la deduzcas de las dos últimas etiquetas. Editar el JSON o el script no cambia la licencia compilada.

Sube los archivos. Añade un script.

Este ejemplo usa una carpeta del mismo origen. El nombre es ilustrativo; utiliza la ruta pública que hayas creado.

1. **Extrae el ZIP.** Sube los dos archivos de ejecución juntos a una nueva carpeta versionada en el área pública del sitio. Conserva sus nombres exactos. Guarda las licencias, los metadatos y el ZIP original en tu archivo de versiones.
2. **Confirma la ruta pública.** Aquí el módulo se sirve en `/assets/charmingviews/v1/accesskit_widget_wasm.js`. Una carpeta del disco, una ubicación del administrador de archivos o un enlace de descarga privado no equivalen a la ruta pública.
3. **Añade este bloque a la plantilla HTML compartida, justo antes de** `</body>`**.** En un CMS, usa su opción admitida de código para todo el sitio que conserve `type="module"`. Guarda y publica la plantilla mediante tu proceso habitual.

```
<script type="module">
  import init, { mount } from
    '/assets/charmingviews/v1/accesskit_widget_wasm.js';
  await init();
  mount();
</script>
```

Copia el bloque completo. Cambia solo la ruta pública de la carpeta. La `/` inicial parte de la raíz del sitio y permite usar los mismos archivos desde páginas anidadas.

Misma carpeta

`accesskit_widget_wasm.js`
`accesskit_widget_wasm_bg.wasm`

Plantilla compartida

Un bloque de módulo por documento

Navegador del visitante

Carga el widget en un dominio autorizado

Por qué aparecen dos llamadas de inicio

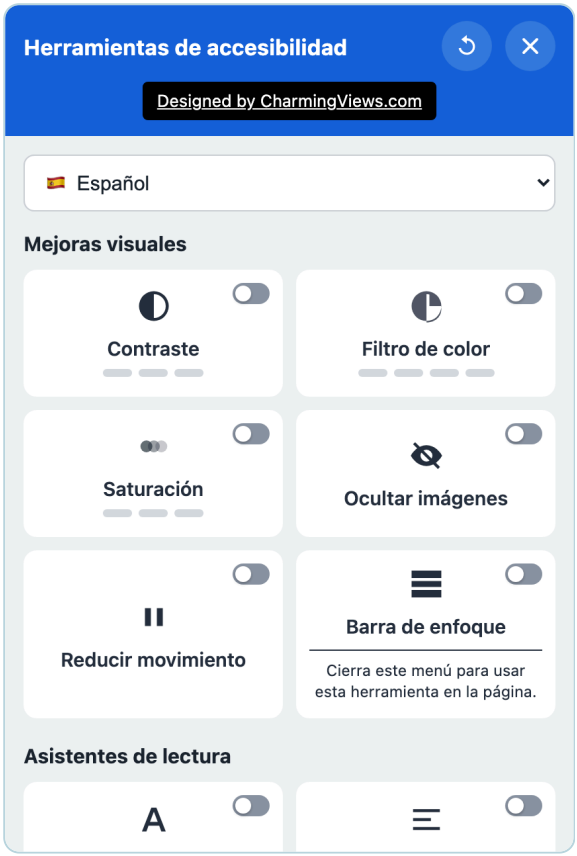
`await init()` carga el WASM correspondiente e inicia el montaje automático cuando el documento está listo. `mount()` es la llamada explícita e idempotente documentada para esa misma instancia del módulo. No implica que debas cargar varias copias.

> **Mantén una sola instalación**
>
> Usa una única integración del widget. Retira un script de AccessKit anterior o coordina una integración existente antes de añadir esta. No lo pegues en cada artículo, componente, cabecera y pie, ni también en un gestor de etiquetas. Los dos archivos suministrados no requieren CSS aparte, clave de licencia, compilador Rust ni instalación con npm.

Compruébalo en una página real

Abre la página de prueba en el dominio autorizado, en una pestaña nueva. Revisa la página publicada, además del editor del CMS.

- [ ] Localiza un solo botón de herramientas de accesibilidad. Ábrelo con el puntero y el teclado; comprueba que siga accesible junto al aviso de cookies o al chat.
- [ ] Aumenta Tamaño del texto un paso. Confirma que cambia el texto de la página y que los controles importantes siguen funcionando. El texto con tamaño fijo en píxeles puede no cambiar.
- [ ] Cierra el panel con Escape. Ábrelo otra vez y usa Restablecer para devolver los ajustes a sus valores iniciales.
- [ ] Cambia una preferencia, recarga la página y comprueba que se conserva. Restablécela después. Las restricciones de almacenamiento o el modo privado pueden afectarla.
- [ ] Visita otra página, una página larga, un formulario y el diseño móvil. Comprueba navegación, foco, desplazamiento, ventanas superpuestas y cualquier proceso de compra.
- [ ] Mantén las comprobaciones existentes de teclado y lector de pantalla. El widget no corrige etiquetas, encabezados, alternativas ni contenido inaccesible.



Compilación real de ejemplo con esquema 3 y panel predeterminado. Las herramientas visibles dependen de la compilación y el navegador.

Si el botón no aparece

Comprueba primero el nombre de host autorizado. Después abre Red y Consola en las herramientas del navegador: ambas peticiones deben completarse sin errores de módulo, WASM o política. El mensaje `AccessKit: disabled on this host` indica un problema de dominio/licencia. Consulta Solución de problemas.

> **Qué se comprobó para esta guía**
>
> Los ejemplos exactos de instalación y configuración se ejecutaron con una compilación comercial real de esquema 3 en Chromium, en un servidor local aislado con el host autorizado dirigido a loopback. Se comprobaron montaje, subdominio, preferencias, restablecimiento, montaje repetido y hosts bloqueados. Esto no garantiza compatibilidad con tu sitio, alojamiento de producción o todos los navegadores.

Ajustes opcionales y sencillos

La instalación predeterminada funciona sin un objeto de configuración. Si necesitas cambios, defínelos *antes* del primer `await init()`.

Este ejemplo mueve el botón a la izquierda, desactiva el atajo del panel y oculta Comandos de voz. **Sustituye al bloque anterior.**

```
<script type="module">
  import init, { mount } from
    '/assets/charmingviews/v1/accesskit_widget_wasm.js';
  window.AccessKitConfig = {
    ui: { side: 'left', hotkey: null },
    features: { voiceControl: false }
  };
  await init();
  mount();
</script>
```

`ui.side`
`'left'` o `'right'`; valor inicial `'right'`. Comprueba posibles solapamientos en pantallas pequeñas.

`ui.hotkey`
Valor inicial `'Alt+Shift+A'`; `null` lo desactiva. Puede entrar en conflicto con atajos del navegador, sistema o tecnología de apoyo.

`features`
Claves booleanas como `voiceControl: false` reducen los controles disponibles. No añaden herramientas excluidas de la compilación.

Idioma de la interfaz

El widget usa el idioma guardado y luego el del navegador, con inglés como alternativa. Su selector cambia la interfaz del widget, no el contenido del sitio. Para seleccionar español desde tu propio control de idioma, ejecuta esto tras la inicialización:

```
if (window.AccessKit) {
  await window.AccessKit.ready;
  window.AccessKit.setLanguage('es');
}
```

Usa `'en'` para inglés. Los códigos admitidos son `en, de, es, it, fr, ru, tr, ar, hi, zh-cn, jp, pt`. Consérvalos exactamente. El idioma de voz y las voces disponibles dependen de otras capacidades del navegador.

Configura una sola vez antes del montaje y recarga por completo. Para cambiar otras dimensiones, colores o traducciones, pide al desarrollador que revise el contrato del código y compruebe usabilidad y contraste.

Alojamiento y seguridad

Para la instalación sencilla, sirve ambos archivos desde el mismo origen HTTPS que la página.

Respuestas HTTP y tipos MIME

Sirve el módulo como JavaScript, por ejemplo `text/javascript`, y el binario como `application/wasm`. Ambas peticiones deben devolver los archivos reales, no un formulario de acceso, una página de error ni la página alternativa de una SPA. El cargador dispone de una alternativa sin streaming para un MIME WASM incorrecto, pero conviene corregir la respuesta. Consulta Referencias del navegador.

El widget comprado no se instala abriendo un archivo local con doble clic. `file://` impone restricciones de origen a los módulos y no tiene un host web autorizado. Esta guía sí se puede leer sin conexión.

Política de seguridad de contenido (CSP)

Si el sitio usa CSP, pide al responsable de seguridad que revise la infracción exacta en Consola. El módulo en línea de esta guía necesita un hash o nonce autorizado, o un archivo de inicio externo aprobado. La carga del módulo, la petición WASM y la compilación WebAssembly también deben cumplir la política. El navegador puede exigir `'wasm-unsafe-eval'` bajo `script-src`.

El widget inserta elementos de estilo y aplica estilos a la página. Una política estricta de `style-src` puede bloquear esas operaciones; el paquete no ofrece un ajuste nonce CSP documentado. Un nonce de script no autoriza por sí solo los estilos generados. Un sitio estricto puede necesitar una integración revisada antes de usar este paquete.

> **Conserva las protecciones del sitio**
>
> No desactives CSP, CORS, protección contra contenido mixto, autenticación ni el cortafuegos web para mostrar el widget. Evita permisos generales `'unsafe-eval'` o comodines. Entrega al desarrollador el error y la petición exactos para que elija el cambio admitido más limitado.

CDN o alojamiento en otro origen

Es una integración opcional para desarrolladores. El servidor de archivos debe permitir los módulos y las peticiones WASM mediante sus respuestas CORS, y la página mediante CSP. El inicializador acepta una URL WASM explícita, pero la pareja y la versión deben coincidir. Usa el ejemplo del mismo origen salvo que exista una razón probada. El host de un CDN no amplía el dominio autorizado de la *página*.

HTTPS y voz

Usa HTTPS en un sitio público y evita recursos HTTP en páginas HTTPS. Comandos de voz depende del reconocimiento del navegador, permiso del micrófono y posiblemente un servicio del proveedor. Revisa esos servicios y tus requisitos de privacidad y consentimiento antes de activar o probar voz.

Tu sitio sigue necesitando revisión

La instalación universal describe un patrón de integración HTML. No garantiza el mismo comportamiento en todos los CMS, diseños o navegadores.

Páginas normales y aplicaciones de página única

Con navegación de página completa, usa la plantilla compartida para que cada documento cargue una vez la misma versión. En una SPA, inicia el widget en la estructura estable de la aplicación, solo en el navegador y cuando exista el DOM. No cargues otra copia en cada ruta, renderizado, hidratación o montaje de componente.

El código mantiene una instancia por módulo y un `mount()` idempotente. No ofrece una API documentada de destrucción o desmontaje. Conserva `#axs-host` fuera de la región que sustituye el framework. URL distintas, incluso con parámetros distintos, pueden crear instancias separadas; usa una URL estable por versión.

Tras cambiar de ruta, revisa los efectos sobre el contenido nuevo, ayudas de lectura y foco, encabezados, formularios y Restablecer. La sustitución del DOM y el CSS pueden afectar al comportamiento. Esta guía comprobó que una navegación del historial conserva un solo botón; no verificó integraciones completas de frameworks específicos.

Marcos, shadow DOM y contenido externo

No supongas que las herramientas alcanzan marcos de otro origen, raíces shadow cerradas o módulos de compra externos. Cada documento tiene sus límites. Comprueba la accesibilidad real de ese contenido en vez de instalar copias adicionales para alcanzarlo.

Límites de móviles y navegadores

- [ ] El visitante necesita un navegador con módulos JavaScript ES y WebAssembly, con scripts permitidos. Esta guía no promete versiones mínimas concretas.
- [ ] Comprueba pantallas pequeñas, orientación horizontal, teclado en pantalla y avisos fijos. La lupa basada en puntero y el cursor grande tienen menos utilidad en dispositivos solo táctiles.
- [ ] La lectura en voz alta usa síntesis de voz del navegador; Comandos de voz usa reconocimiento y permisos. Pueden no estar disponibles. La lectura según el foco no sustituye un lector de pantalla nativo.
- [ ] La separación de palabras depende del idioma del documento y del navegador. El idioma del widget no traduce contenido ni añade un idioma de documento ausente.
- [ ] Las preferencias usan el almacenamiento local del origen cuando está disponible. No se sincronizan entre navegadores, dispositivos ni orígenes de subdominios.

> **Mantén el contenido accesible desde su origen**
>
> El widget ofrece preferencias y ayudas al visitante. No repara automáticamente el contenido, certifica cumplimiento WCAG/ADA, sustituye HTML semántico ni garantiza eliminar todas las barreras de accesibilidad.

Actualizar, revertir o retirar

Actualiza sin mezclar versiones

1. **Conserva la versión que funciona.** Archiva ambos archivos, metadatos, licencias, bloque y sumas del paquete. Anota la ruta pública y la fecha.
2. **Comprueba el paquete nuevo.** Verifica base y perfil. Compara los archivos con `checksums.json` mediante tus herramientas de publicación. Las sumas detectan cambios; no demuestran por sí solas quién suministró el paquete.
3. **Sube a una carpeta nueva.** Por ejemplo, `/assets/charmingviews/v2/`. Termina de subir ambos archivos antes de cambiar la ruta en la plantilla. `v1`/`v2` son nombres de tus carpetas, no versiones prometidas del producto.
4. **Prueba y cambia el bloque compartido.** Repite las comprobaciones de Compruébalo en una página real. Conserva una sola instalación. Hace falta una recarga completa; una página ya montada no cambia de código automáticamente.
5. **Revisa las cachés.** Confirma las peticiones a la carpeta nueva. Revisa CMS/CDN y service worker si persiste contenido antiguo. Un parámetro solo en la URL del JS no cambia también la URL del WASM.

Volver a la versión anterior

Restaura la ruta anterior en la plantilla compartida, apuntando a la pareja archivada. Actualiza o invalida las cachés HTML correspondientes, recarga y verifica. Conserva la versión fallida para investigar; no mezcles su JS con el WASM anterior.

Desinstalación limpia

1. Opcionalmente, usa Restablecer antes de retirar el widget. Elimina preferencias conocidas de herramientas, pero conserva el idioma guardado.
2. Elimina el bloque y todo código propio que llame a `window.AccessKit`. Retira también cualquier integración antigua del gestor de etiquetas que cargue el widget.
3. Publica la plantilla, actualiza las cachés HTML/service worker y recarga por completo. Confirma que desaparece el botón y ya no se solicitan los archivos.
4. Retira la carpeta de ejecución cuando ninguna página dependa de ella. Conserva el ZIP comprado y la licencia en tus archivos.

> **No borres todo el almacenamiento del sitio**
> Al retirar el script se impide el montaje futuro tras una navegación o recarga completa. No se borran las preferencias del visitante ni se desmonta de inmediato una página abierta. Para una prueba local limpia, un desarrollador puede eliminar solo las claves de preferencias conocidas `axs.*`, incluido `axs.lang` si lo desea. No uses `localStorage.clear()`: podría borrar datos ajenos al widget.

La compilación comercial de esquema 3 no caduca por tiempo durante la ejecución. Eso describe el binario; no promete futuras actualizaciones, duración del soporte, transferencias ni condiciones adicionales de licencia.

Solución de problemas

Empieza por el síntoma y comprueba la respuesta o el host antes de cambiar código.

No hay botón; Consola dice “disabled on this host”

Compara el host de la página con `licensedBaseDomain` en `build-configuration.json`. Usa un subdominio de prueba bajo esa base. No se incluyen localhost, IP ni vistas previas del proveedor. Si la base suministrada es incorrecta, comunícalo con tu compra por la vía de soporte/contacto de tu pedido; no edites ni parches el binario.

“Failed to load module script”, error MIME JavaScript o 404

Comprueba la URL exacta en Red. Debe devolver el JS y un MIME JavaScript. Corrige ruta pública, mayúsculas, permisos o respuesta del servidor. Revisa si el CMS eliminó `type="module"` o mostró el código como texto.

Error al descargar, compilar o enlazar WASM, o de “magic word”

Revisa la respuesta WASM. Puede ser una página de acceso/error, transferencia incompleta o pareja incompatible. Vuelve a subir ambos archivos del mismo ZIP y verifica sus sumas. Sirve `application/wasm`. Revisa por separado los errores CSP.

Mensaje de CSP, CORS o contenido mixto

Entrega al administrador del sitio/seguridad la URL y directiva bloqueadas. Prefiere HTTPS del mismo origen. Sigue Alojamiento y seguridad; no desactives las protecciones.

Dos botones, acciones repetidas o paneles en conflicto

Busca todos los cargadores en plantilla, etiquetas, plugins y componentes. Conserva una ruta de módulo y una integración. Elimina duplicados y recarga. Repetir `mount()` en un módulo es idempotente; cargar copias separadas es distinto.

El botón aparece pero no cambia parte de la página

Revisa CSS fijo en píxeles, contenido sustituido, marcos, límites shadow y otros widgets. Prueba una herramienta cada vez y usa Restablecer. El widget no reescribe contenido de origen inaccesible.

> **Recoge información útil**
>
> Anota la versión de guía/paquete o identidad de código, ruta afectada, navegador y versión, dispositivo, pasos y primer error relevante de Consola/Red. Oculta tokens, URL privadas, valores de formularios y datos personales. No compartas un registro completo de red sin revisarlo.

Más comprobaciones y referencias

Las preferencias no se guardan

Revisa modo privado, restricciones y cambios de origen. `www` y raíz pueden compartir licencia y mantener almacenamiento separado. Prueba tras recargar por completo.

Ajustes, idioma o lista de herramientas incorrectos

Configura antes de inicializar y recarga. El idioma guardado puede prevalecer sobre el navegador. La compilación, configuración o capacidades del navegador pueden limitar herramientas. Una opción no añade una herramienta no compilada.

No funciona la voz o el micrófono

Comprueba compatibilidad, permisos, HTTPS y servicios del proveedor. No solicites micrófono solo para verificar la instalación básica. Usa los demás controles cuando no haya voz.

Sigue cargando la versión anterior

Revisa URL reales, cabeceras/cachés y service worker. Cambia ambos archivos mediante otra carpeta versionada y actualiza la caché HTML. Una recarga forzada puede no renovar la caché de un service worker.

Diagnóstico opcional de solo lectura

Un desarrollador puede ejecutarlo en Consola en la página publicada afectada. La ausencia de API puede significar inicio fallido o host rechazado; combínalo con los errores y las respuestas de Red.

```
const kit = window.AccessKit;
console.table({
  hostname: location.hostname,
  apiPresent: Boolean(kit),
  enabled: kit?.enabled,
  language: kit?.lang
});
if (kit) console.log(kit.buildConfig);
```

Referencias del navegador

Las instrucciones y ejemplos esenciales están incluidos para leer sin conexión. Estas referencias opcionales en línea explican reglas del navegador; no prometen compatibilidad ni soporte del widget.

- MDN: módulos JavaScript (inglés)—MIME, scripts de módulo y archivos locales.
- MDN: WebAssembly.instantiateStreaming() (inglés)—MIME de streaming y políticas.
- MDN: CSP script-src (inglés)—autorización de scripts y compilación WebAssembly.
- MDN: CORS (inglés)—peticiones entre orígenes distintos.

Si alguna herramienta no funciona como esperas en una página concreta, envíanos la URL de la página y describe el problema. Haremos lo posible por investigarlo y ayudarte, aunque no podemos garantizar una solución para todos los sitios web.

Contacto: support@buzzforus.com.

Conserva esta guía con el paquete. El uso se rige por `LICENSE.txt`. Sigue las instrucciones de soporte/contacto de tu compra; esta guía no establece plazos de respuesta ni políticas de reembolso.