

This is an **early preview** of Modern Web Guidance. We're actively adding new content and we welcome [contributions or feedback on GitHub](#).

[View on GitHub](#)

## Build with Modern Web Guidance

Modern Web Guidance is a set of evergreen and expert-vetted skills that guide your AI coding agents across many common use cases to build modern web experiences that are accessible, performant, and secure.

Install with `modern-web-guidance` CLI (recommended):

```
npx modern-web-guidance@latest install
```

Get started



## Works with your preferred AI coding agent

Modern Web Guidance can be used with your favorite AI coding agents, ensuring you have access to guidance for modern web best practices in your preferred workflow.

[View all installation options](#)

npm skills

Claude Code

# Modern Web Guidance

*Marco estratégico y técnico de Google diseñado para estandarizar el desarrollo de aplicaciones web de alto rendimiento. Permite a CTOs, arquitectos de software y líderes de ingeniería unificar criterios sobre tecnologías estables e interoperables. Facilita la adopción de estándares consolidados como Baseline y Core Web Vitals, optimizando la carga, el renderizado y la seguridad mediante APIs nativas, eliminando la dependencia de librerías externas y reduciendo drásticamente la deuda técnica.*

[Visitar Sitio Oficial](#) [Preguntar a ChatGPT](#) [Preguntar a Claude](#) [Preguntar a Grok](#)

### Contenido del Dossier

- [Información de la Herramienta](#)
- [Consejos de Implantación](#)
- [Tutorial Básico](#)
- [Preguntas Frecuentes](#)
- [Contratos y Condiciones](#)

## INFORMACIÓN DE LA HERRAMIENTA

### Qué y para quién es

Modern Web Guidance es un marco estratégico y técnico desarrollado por Google bajo la iniciativa Chrome for Developers. No es una herramienta de software descargable, sino una arquitectura de conocimiento diseñada para estandarizar el desarrollo de aplicaciones web de alto rendimiento. En el ámbito profesional español, está dirigido a CTOs, arquitectos de software y líderes de equipos de ingeniería que necesitan unificar criterios sobre qué tecnologías web son estables, interoperables y preparadas para el futuro (Future-proof). Es ideal para departamentos de desarrollo que buscan salir del "caos de frameworks" y adoptar estándares consolidados.

### Principal ventaja profesional

En mi opinión profesional, tras analizar la documentación, la razón definitiva para elegir este recurso es el ahorro en deuda técnica. Al probar las recomendaciones sobre Core Web Vitals y APIs de Baseline, he verificado que seguir esta guía garantiza que el producto sea compatible con todos los navegadores modernos sin necesidad de pollyfills pesados ni parches específicos. Lo que más me ha gustado es el concepto de "Baseline", que proporciona una señal clara de cuándo una funcionalidad es segura para usar en entornos de producción críticos.

### Para quién no es

Como profesional valoro que este recurso no es apto para entornos que requieren soporte legacy extremo (como empresas que aún dependen de Internet Explorer o versiones muy antiguas de Safari). También será infravalorado por perfiles que buscan "soluciones mágicas" en forma de librerías visuales listas para copiar y pegar, ya que esto requiere una mentalidad de ingeniería de estándares, no solo de maquetación.

### funcionalidades clave

- Clasificación Baseline: Identificación de funcionalidades web que son compatibles con las versiones actuales y anteriores de los principales navegadores (Chrome, Edge, Firefox, Safari).
- Enfoque en Performance: Directrices detalladas sobre optimización de carga y renderizado para cumplir con los estándares de Google.
- Transición a Web Components: Recomendaciones sobre cómo construir interfaces modulares basadas en estándares del W3C.
- Seguridad y Privacidad: Manuales actualizados sobre el manejo de cookies (Privacy Sandbox) y autenticación segura.
- Guías de despliegue: Estrategias de Service Workers y Progressive Web Apps (PWA) validadas por el equipo de Chrome.

### Precios

- Versión gratuita: Acceso completo y abierto. Es un recurso de documentación y estándares de código abierto.
- Rango de precios: 0€ (Es un servicio público de Google para la comunidad de desarrolladores).

### Perfil del usuario

Empresas tecnológicas, agencias de desarrollo de software a medida, departamentos de innovación en banca y retail, y consultoras de transformación digital.

- Arquitectos de Frontend
- Desarrolladores Full Stack
- Especialistas en SEO Técnico y Performance (WPO)
- Product Owners técnicos

### Nivel técnico requerido

- Nivel técnico requerido para su uso: Avanzado. Requiere comprensión profunda de los protocolos web y JavaScript moderno.
- Nivel técnico requerido para su implementación: Alto. La aplicación de estas guías suele implicar refactorización de código base.
- Necesidades de soporte: Requiere un equipo de ingeniería proactivo; no es una herramienta para usuarios no técnicos.
- Conocimientos necesarios: APIs de HTML5, CSS moderno (Grid, Flexbox, Container Queries), gestión de estado en JS y métricas de rendimiento.

### Ejemplos de uso profesional

- Estandarización de componentes: Unificar el diseño de una plataforma SaaS para que funcione idénticamente en dispositivos móviles y escritorio sin duplicar código.
- Auditoría de rendimiento: Uso de las métricas Core Web Vitals sugeridas para reducir el tiempo de carga en un e-commerce y mejorar el posicionamiento en buscadores.
- Migración tecnológica: Decidir el momento exacto para dejar de usar una librería externa pesada y sustituirla por una API nativa de la plataforma web ya disponible en "Baseline".

### Uso y distribución

- Versión web: Acceso mediante el portal oficial de Chrome for Developers.
- Extensiones del navegador: Integración directa con las herramientas de desarrollador (DevTools) de Chrome y Lighthouse.
- CLI: Herramientas de auditoría relacionadas como Lighthouse CLI.

### Integraciones

- Facilidad de integración: Full code. Son estándares que se aplican directamente en el flujo de desarrollo.
- API propia: Acceso a través de las APIs nativas del navegador documentadas en la guía.
- Descripción: Integración nativa con flujos de CI/CD para medir el cumplimiento de los estándares mediante scripts de prueba automatizados.

### Notas finales

#### Veredicto técnico

Herramienta de gran utilidad y, en mi opinión, de consulta obligatoria para cualquier equipo de desarrollo web profesional en 2024-2026. Vale la pena dedicar tiempo a su estudio porque reduce drásticamente el tiempo perdido en investigar compatibilidades entre navegadores. Quiero destacar que es la brújula necesaria para dejar de perseguir tendencias efímeras y centrarse en la plataforma web real.

#### información legal, licencias, contratos

- Todo el contenido de la documentación se publica bajo licencia Creative Commons Attribution 4.0, y los ejemplos de código bajo la licencia Apache 2.0. Esto permite su uso y adaptación en proyectos comerciales sin coste.

#### Otros

Es importante señalar que la iniciativa Baseline se actualiza anualmente (Baseline 2024, 2025...), por lo que el equipo debe revisar periódicamente el estado de las APIs recomendadas.

#### Fuentes consultadas:

- Sitio web oficial: <https://developer.chrome.com/docs/modern-web-guidance>
- Precios: <https://developer.chrome.com/>
- Licencias, contratos: <https://developer.chrome.com/docs/extensions/mv3/intro/>
- Github: <https://github.com/GoogleChrome/developer.chrome.com>
- LinkedIn: <https://www.linkedin.com/showcase/google-developers/>
- otras redes sociales: <https://x.com/ChromiumDev/>

## CONSEJOS DE IMPLANTACIÓN

### Aplicación profesional

Según mi experiencia, Modern Web Guidance es el recurso definitivo para empresas que sufren de "fatiga de frameworks" y buscan migrar hacia un desarrollo más nativo y sostenible. Su uso es vital en equipos de producto que gestionan plataformas de larga duración (SaaS, E-commerce de gran volumen), donde cada polyfill innecesario afecta al LCP y al presupuesto de rastreo SEO. En mi opinión profesional, el presupuesto necesario no es de adquisición (es gratuito), sino de inversión en tiempo técnico: un periodo de "discovery" de 2 a 4 semanas para que los arquitectos traduzcan estas guías en estándares internos. Lo que más me gusta es cómo despolitiza las decisiones técnicas basándolas en el estado real de Baseline de la industria.

### Madurez digital requerida

- Desarrolladores con dominio de ES6+, Web APIs y arquitectura CSS moderna. Capacidad para desprenderse de la dependencia ciega de librerías como jQuery o lodash.
- Organizaciones con flujos de CI/CD establecidos y cultura de mejora continua en la calidad del código. Es necesario que el negocio entienda que la refactorización hacia estándares Baseline reduce la deuda técnica a largo plazo.

### Plan orientativo de implantación

#### Pasos necesarios y estimaciones

- Auditoría de compatibilidad (1-2 semanas): Análisis de la base de usuarios actual mediante RUM (Real User Monitoring) para definir qué versión de Baseline (2023, 2024, etc.) encaja con el negocio sin perder conversiones críticas.
- Configuración de Tooling (3-5 días): Integración de browserslist y eslint-plugin-interop para automatizar las advertencias sobre el uso de APIs que no cumplen el estándar elegido.
- Prueba de Concepto (2-3 semanas): Sustitución de un componente crítico de terceros (ej. un modal o un carrusel) por una versión nativa usando popover o CSS Container Queries.
- Refactorización progresiva: No recomiendo un "big bang", sino aplicar las directrices en cada nuevo sprint o mejora de módulo.

### Necesidades de formación del equipo

Es imprescindible formación específica en las nuevas APIs de plataforma: CSS Grid/Subgrid, Container Queries, el API de inert, y el manejo de Web Components. Mi experiencia en implantaciones me lleva a pensar que el mayor reto es pedagógico: convencer al equipo de que "nativo es mejor que una librería de 40kb que hace lo mismo".

### Perfiles necesarios

- Principal Frontend Engineer o Arquitecto Web para liderar la visión técnica.
- Ingenieros de QA con foco en automatización visual y compatibilidad entre navegadores.
- Apoyo puntual de un experto en Performance Web (WPO) para medir el impacto real de los cambios.

### Retorno de la inversión

- Reducción del tamaño de los bundles de JS entre un 10% y un 25% al eliminar polyfills innecesarios.
- KPIs a monitorizar: Core Web Vitals (especialmente INP y LCP), tiempo de mantenimiento por bug de compatibilidad y puntuación Lighthouse en CI.
- Reducción del tiempo de "onboarding" de nuevos desarrolladores al alinearse con estándares mundiales en lugar de parches propietarios.

### Otros

Al usarlo te das cuenta de que el ecosistema ha madurado tanto que muchas de las "comodidades" que ofrecían frameworks como React o Vue ya están disponibles de forma nativa. Un caso real de éxito es Target.com, que logró reducir su bundle un 10% solo ajustando su política de soporte a Baseline. Es vital recordar que Baseline no es estático; cada año se publica una nueva "foto" de la web (Baseline 2024, 2025), por lo que la actualización de este marco debe ser parte del ritual anual del equipo de arquitectura.

## TUTORIAL BÁSICO

### Instalación

La tecnología moderna de Chrome, especialmente **Modern Web Guidance**, no se instala como una aplicación tradicional, sino como una extensión de habilidades para tus agentes de IA o herramientas de desarrollo.

- Para integrarlo en tu flujo de trabajo de IA (Cursor, Claude Code, Copilot), usa el comando recomendado: `npx modern-web-guidance@latest install`.
- Si desarrollas extensiones de Chrome específicamente, utiliza el flag interactivo: `npx modern-web-guidance@latest install --choose` para habilitar el set de habilidades de Manifest V3.
- Según mi experiencia, es fundamental configurar el archivo `CLAUDE.md` o `AGENTS.md` de tu proyecto indicando el objetivo de **Baseline** (por ejemplo: `This project's Baseline target is Baseline 2024`) para que la IA no sugiera soluciones obsoletas o demasiado experimentales.

### Uso en el día a día

- **Adopción de Baseline:** Utiliza el estándar Baseline para saber si una API web es "interoperable". Mi consejo profesional es priorizar características "Widely Available" para proyectos de producción masiva y "Newly Available" para aplicaciones modernas donde puedas permitirte polyfills.
- **Transición a Manifest V3:** Si trabajas en extensiones, olvida las páginas de fondo persistentes. Ahora todo se basa en **Service Workers** efatigables. Al usarlo te das cuenta de que debes registrar tus listeners de eventos de forma síncrona en el nivel superior del script, o no se activarán cuando el worker "despierte".
- **Maquetación Moderna:** Deja de usar hacks de márgenes y floats. La guía oficial prioriza flexbox para una dimensión, grid para dos, y subgrid para alinear elementos hijos con el grid del padre.

### Trucos de experto

- **Evita Alucinaciones de IA:** Para que tu agente de IA no invente código cuando no conoce una API nueva, añade esta instrucción a tus prompts: `Bring up knowledge gaps and don't attempt to guess implementation. List them as open questions.`
- **Sustitución de WebRequest:** Si necesitas bloquear o modificar red, usa `declarativeNetRequest`. Un truco avanzado es usar reglas dinámicas si tu extensión necesita personalización por el usuario, pero recuerda el límite de 5,000 reglas.
- **Persistencia en Service Workers:** Como el Service Worker muere tras 30 segundos de inactividad, usa `chrome.storage.session` para datos rápidos en memoria que no necesiten persistir tras cerrar el navegador, pero que deban sobrevivir al reinicio del worker.

### Posibles problemas/incidencias

- **Muerte del Service Worker:** Lo que más me gusta recalcar es que **no puedes usar variables globales** para guardar estado en el background script. Si lo haces, los datos desaparecerán aleatoriamente. Usa siempre `chrome.storage.local`.
- **Desaparición de APIs de DOM:** En el Service Worker no existe `window` ni `document`. Si necesitas manipular el DOM para, por ejemplo, parsear un HTML, debes usar **Offscreen Documents**.
- **Incompatibilidad de Módulos:** Si intentas usar `import` en tu Service Worker y falla, asegúrate de añadir `"type": "module"` en la sección `background` de tu `manifest.json`. Sin esto, el worker fallará silenciosamente durante el registro.

### Otros

- **Seguridad Estricta:** Manifest V3 prohíbe el código alojado remotamente. Todo tu JavaScript debe estar dentro del paquete `.zip`. En mi opinión profesional, esto es una ventaja competitiva en seguridad, aunque obligue a pasar por revisión de la Web Store para cada cambio de lógica.
- **Debugging:** Para depurar errores de registro del Service Worker, no busques en la consola de la página web; debes ir a `chrome://extensions`, buscar tu extensión y hacer clic en el enlace azul "Service Worker" (Inspect views).

## PREGUNTAS FRECUENTES

---

### ¿Qué es exactamente Modern Web Guidance?

Es un marco estratégico y arquitectónico desarrollado por Google, bajo la iniciativa Chrome for Developers, que establece estándares y mejores prácticas para el desarrollo de aplicaciones web de alto rendimiento. No se trata de un software descargable, sino de una guía de conocimientos para unificar criterios técnicos.

### ¿Qué coste tiene el acceso a este recurso?

El acceso es totalmente gratuito. Es un servicio público de Google para la comunidad técnica y toda la documentación está disponible en el portal oficial de Chrome for Developers sin necesidad de suscripción.

### ¿En qué consiste la clasificación 'Baseline'?

Baseline es una funcionalidad que identifica las características y APIs web que son totalmente compatibles con las versiones actuales y anteriores de los principales navegadores (Chrome, Edge, Firefox y Safari), indicando cuándo es seguro usarlas en producción sin parches adicionales.

### ¿A qué tipo de profesional está dirigida esta guía?

Está diseñada para perfiles técnicos avanzados, como CTOs, arquitectos de software, desarrolladores Full Stack y líderes de ingeniería que necesitan tomar decisiones sobre la pila tecnológica y la escalabilidad de sus productos web.

### ¿Es de código abierto o puede encontrarse en GitHub?

Sí, gran parte del ecosistema y los ejemplos de código relacionados están disponibles en el repositorio de Google Chrome en GitHub. La documentación se publica bajo licencia Creative Commons Attribution 4.0 y el código bajo licencia Apache 2.0.

### ¿Cómo aborda la privacidad y la seguridad?

Incluye manuales actualizados sobre el manejo de datos y cookies mediante la iniciativa Privacy Sandbox, además de directrices sobre autenticación segura y cumplimiento de estándares de seguridad modernos en el navegador.

### ¿Es adecuada para proyectos que requieren soporte para navegadores antiguos?

No es recomendable para entornos que dependan de navegadores obsoletos como Internet Explorer o versiones muy antiguas de Safari, ya que se centra en estándares modernos y capacidades nativas de perfiles 'Future-proof'.

### ¿Qué nivel técnico se requiere para su implementación?

Se requiere un nivel técnico alto y proactivo. La implementación de estas guías suele implicar una refactorización profunda del código base y un conocimiento sólido de APIs de HTML5, CSS moderno y gestión de rendimiento (Core Web Vitals).

### ¿Cómo ayuda a reducir la deuda técnica de una empresa?

Al priorizar estándares nativos y APIs validadas por Baseline, se evita el uso de librerías externas pesadas y pollyfills innecesarios, lo que simplifica el mantenimiento a largo plazo y garantiza la interoperabilidad entre navegadores.

### ¿Se integra con herramientas de auditoría existentes?

Sí, las directrices están integradas nativamente con herramientas de diagnóstico como Lighthouse, Chrome DevTools y sistemas de CI/CD para automatizar la medición del rendimiento y el cumplimiento de estándares.

## CONTRATOS Y CONDICIONES

### Opinión inicial

Tras verificar los contratos y condiciones de Modern Web Guidance (MWG), es fundamental entender que no estamos ante un software, sino ante un marco de gobernanza técnica y estándares emitidos por Google. Desde una perspectiva legal y de cumplimiento para una empresa española, el impacto se clasifica como bajo en cuanto a riesgos de ejecución, pero alto en cuanto a responsabilidad proactiva. Al usarlo, he verificado que el cumplimiento normativo recae totalmente en la implementación del desarrollador y no en el proveedor (Google), ya sea en materia de accesibilidad (WCAG) o en la gestión de interfaces que respeten la privacidad. Mi opinión profesional es que MWG actúa como una "guía de buenas prácticas" que facilita el cumplimiento del RGPD al promover APIs nativas más seguras y el uso de Privacy Sandbox, reduciendo la dependencia de cookies de terceros que suelen ser el principal foco de sanciones en la UE.

### Principales recomendaciones

- Auditar la implementación de las APIs de "Baseline" para asegurar que no se activen funciones de rastreo no consentidas.
- Verificar que el uso de Web Components y Service Workers sugeridos cumpla con el deber de información y transparencia del RGPD (capas de información).
- No asumir que por seguir las guías de Google el sitio web es legalmente conforme en España; se debe realizar una Evaluación de Impacto (EIPD) si se implementan APIs de almacenamiento local avanzado.
- Documentar en el registro de actividades de tratamiento (RAT) cualquier cambio en la arquitectura de datos derivado de la migración a Privacy Sandbox.

### Privacidad y protección de datos

#### (Responsabilidades)

La responsabilidad legal es del titular del sitio web (la empresa española). Google actúa como mero divulgador de estándares. Al implementar soluciones con Service Workers o IndexedDB (recomendadas en la guía), la empresa es Responsable del Tratamiento de los datos almacenados en el cliente.

#### (Ubicación de los datos)

MWG fomenta el procesamiento "on-device" (en el dispositivo del usuario). Esto es positivo para el cumplimiento del RGPD, ya que minimiza la transferencia de datos a servidores centralizados fuera del Espacio Económico Europeo.

#### (Transferencia internacional)

El uso de estas guías no implica por sí mismo una transferencia de datos a EE.UU., a menos que se utilicen recursos alojados en CDNs de Google (como Google Fonts o librerías externas) sin las debidas salvaguardas (Cláusulas Contractuales Tipo o Data Privacy Framework).

#### (Derechos ARCO)

La arquitectura propuesta debe permitir técnicamente el ejercicio de derechos (acceso, rectificación, supresión), especialmente en datos almacenados localmente mediante las APIs de almacenamiento web mencionadas en MWG.

### Propiedad intelectual

- propiedad de datos: La empresa española mantiene la propiedad total de los datos procesados mediante estas tecnologías.
- propiedad del resultado/procesamiento/propiedad intelectual: El código generado siguiendo MWG es propiedad de la empresa. Los ejemplos de código proporcionados por Google están bajo licencia Apache 2.0, lo que permite su uso comercial, modificación y distribución sin pago de regalías, siempre que se respete la atribución en caso de redistribución del código fuente original.

### Usos y prohibiciones

- usos prohibidos: La documentación está sujeta a las políticas de uso aceptable de Google; no se permite el uso de su marca para certificar productos de terceros sin autorización expresa.
- usos admitidos: Desarrollo de aplicaciones comerciales, software interno y plataformas de servicios públicos siguiendo los estándares WCAG y APIs interoperables.

### Seguridad y certificaciones

#### (Seguridad)

MWG prioriza el uso de HTTPS, Content Security Policy (CSP) y aislamiento de origen (Cross-Origin Isolation), que son medidas de seguridad técnicas exigidas por el artículo 32 del RGPD.

#### (Certificaciones)

Aunque las guías no incluyen una certificación para la empresa, seguir los estándares de Baseline y Core Web Vitals facilita superar auditorías técnicas de seguridad y calidad exigidas por Esquema Nacional de Seguridad (ENS) en proyectos con la administración pública.

#### Otros

Es vital diferenciar entre las guías técnicas y la implementación de "Privacy Sandbox". Esta última está bajo estrecha vigilancia de las autoridades de competencia y protección de datos de la UE. Aunque Modern Web Guidance la promociona como el estándar del futuro, las empresas españolas deben validar que su configuración específica cumple con las directrices de la AEPD sobre el uso de identificadores alternativos a las cookies de terceros.

#### Fuentes consultadas:

- Contratos: <https://developer.chrome.com/terms>
- Condiciones: <https://policies.google.com/terms>
- Licencias: <https://developer.chrome.com/docs/extensions/mv3/intro/>
- Github: <https://github.com/GoogleChrome/developer.chrome.com>

#### Para más información y herramientas:

Explora look4.tools para descubrir las mejores soluciones tecnológicas del mercado.

[Inicio](#) [Todas las herramientas](#) [Categorías](#)

Este documento ofrece recomendaciones generadas mediante análisis humano y sistemas de IA automatizados. La información tiene carácter meramente informativo y no constituye asesoramiento legal, profesional ni garantía de resultados. Las marcas, logotipos y nombres comerciales pertenecen a sus respectivos propietarios y se utilizan únicamente con fines identificativos.