Juan Muñoz

Fundador de Root

+18 años de experiencia
en plataformas con base web.

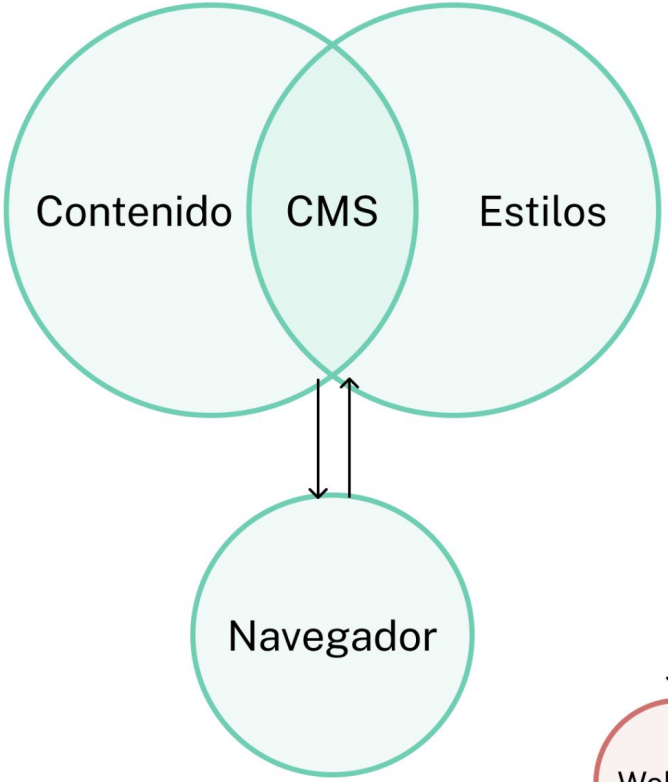
r/oot

headless-agency

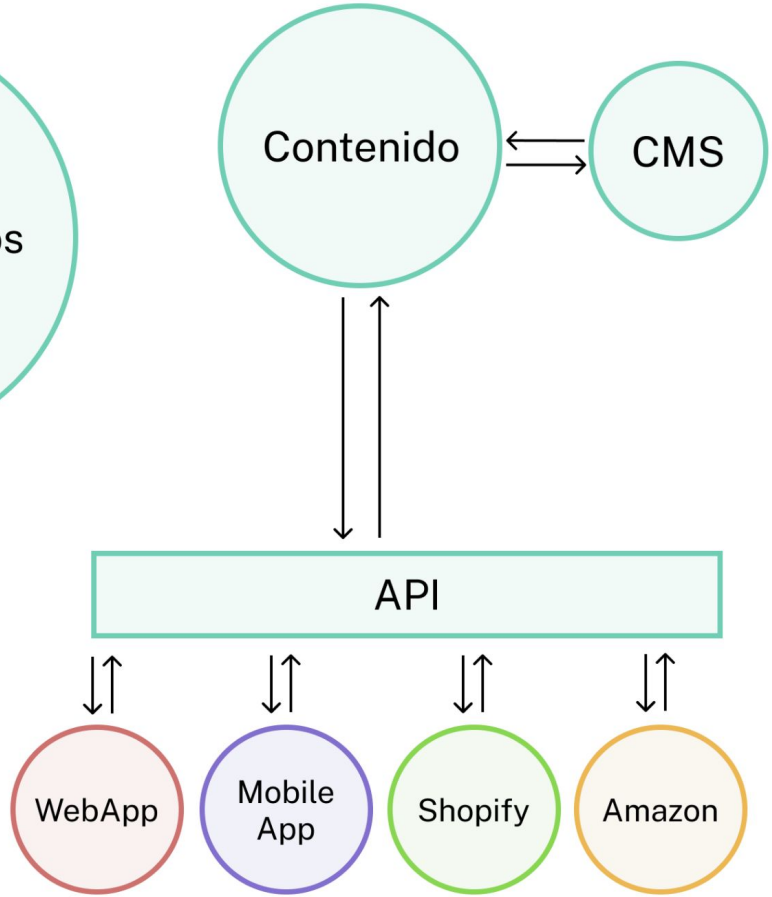
¿Qué es un CMS *headless*?

El nombre *headless* viene de la **separación de la cabeza *o* front-end, y el cuerpo o back-end** del sistema.

CMS Tradicional

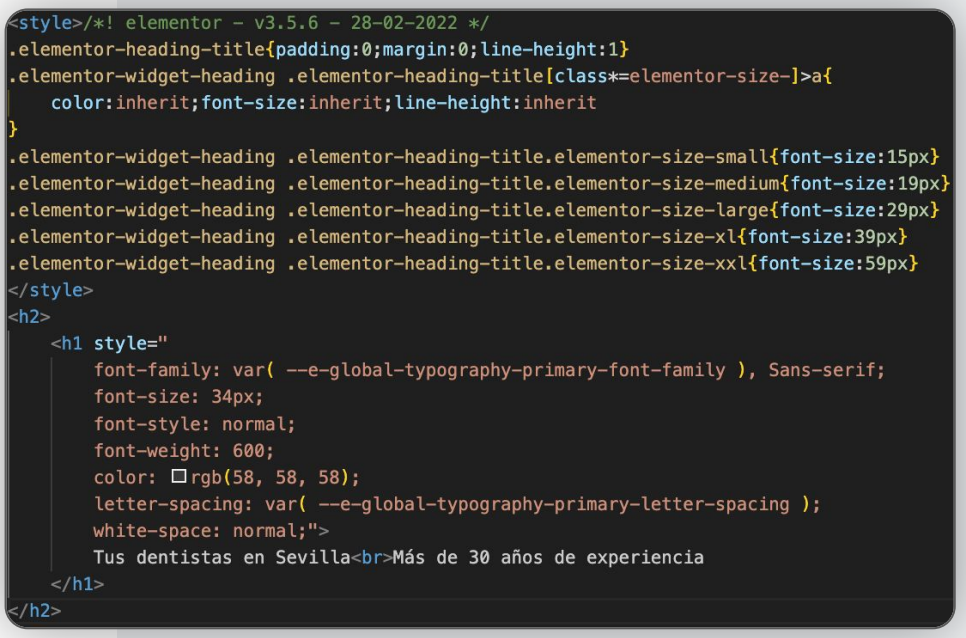


CMS Headless



Repaso de los **sistema tradicionales**

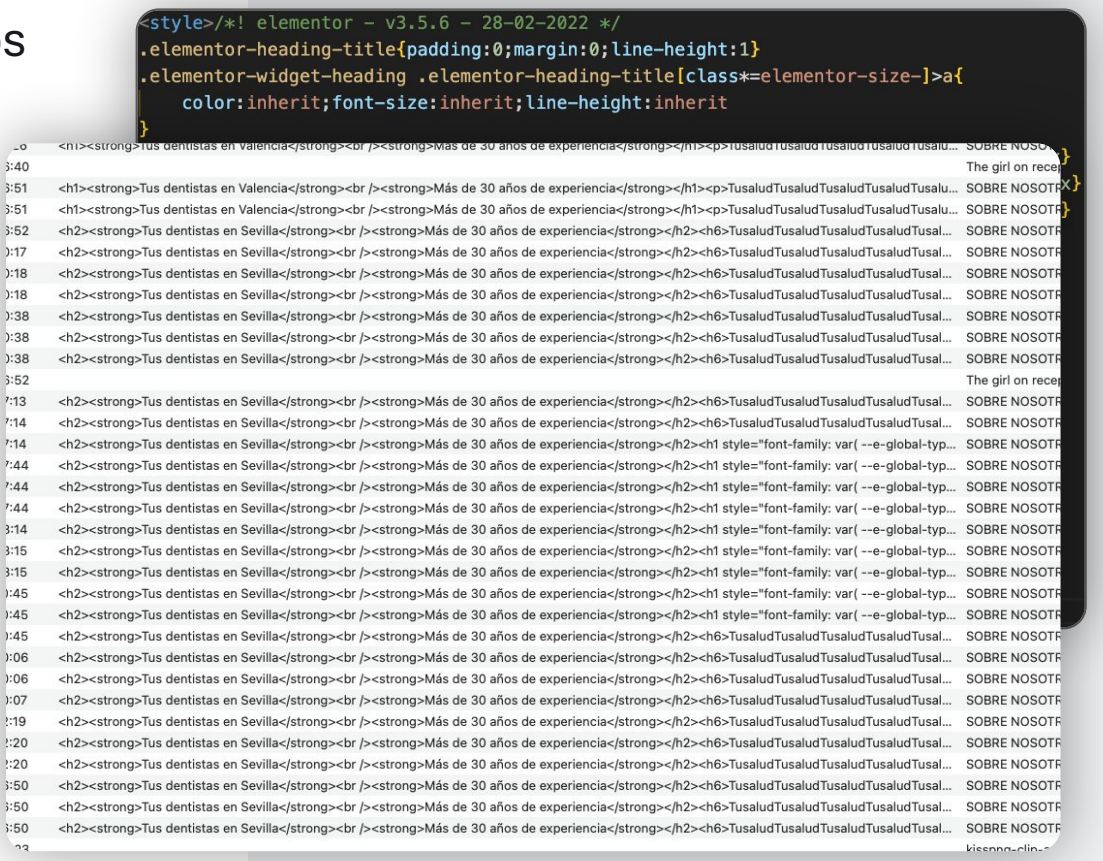
Sistemas como Wordpress, **almacenan información de estilos y layout** en base de datos.



```
<style>/*! elementor - v3.5.6 - 28-02-2022 */
.elementor-heading-title{padding:0;margin:0;line-height:1}
.elementor-widget-heading .elementor-heading-title[class*=elementor-size-]>a{
    color:inherit;font-size:inherit;line-height:inherit
}
.elementor-widget-heading .elementor-heading-title.elementor-size-small{font-size:15px}
.elementor-widget-heading .elementor-heading-title.elementor-size-medium{font-size:19px}
.elementor-widget-heading .elementor-heading-title.elementor-size-large{font-size:29px}
.elementor-widget-heading .elementor-heading-title.elementor-size-xl{font-size:39px}
.elementor-widget-heading .elementor-heading-title.elementor-size-xxl{font-size:59px}
</style>
<h2>
    <h1 style="
        font-family: var( --e-global-typography-primary-font-family ), Sans-serif;
        font-size: 34px;
        font-style: normal;
        font-weight: 600;
        color: rgb(58, 58, 58);
        letter-spacing: var( --e-global-typography-primary-letter-spacing );
        white-space: normal;">
        Tus dentistas en Sevilla<br>Más de 30 años de experiencia
    </h1>
</h2>
```

Así es como se verían los datos guardados en tablas

El código anterior se guarda como **texto plano** en la tabla.

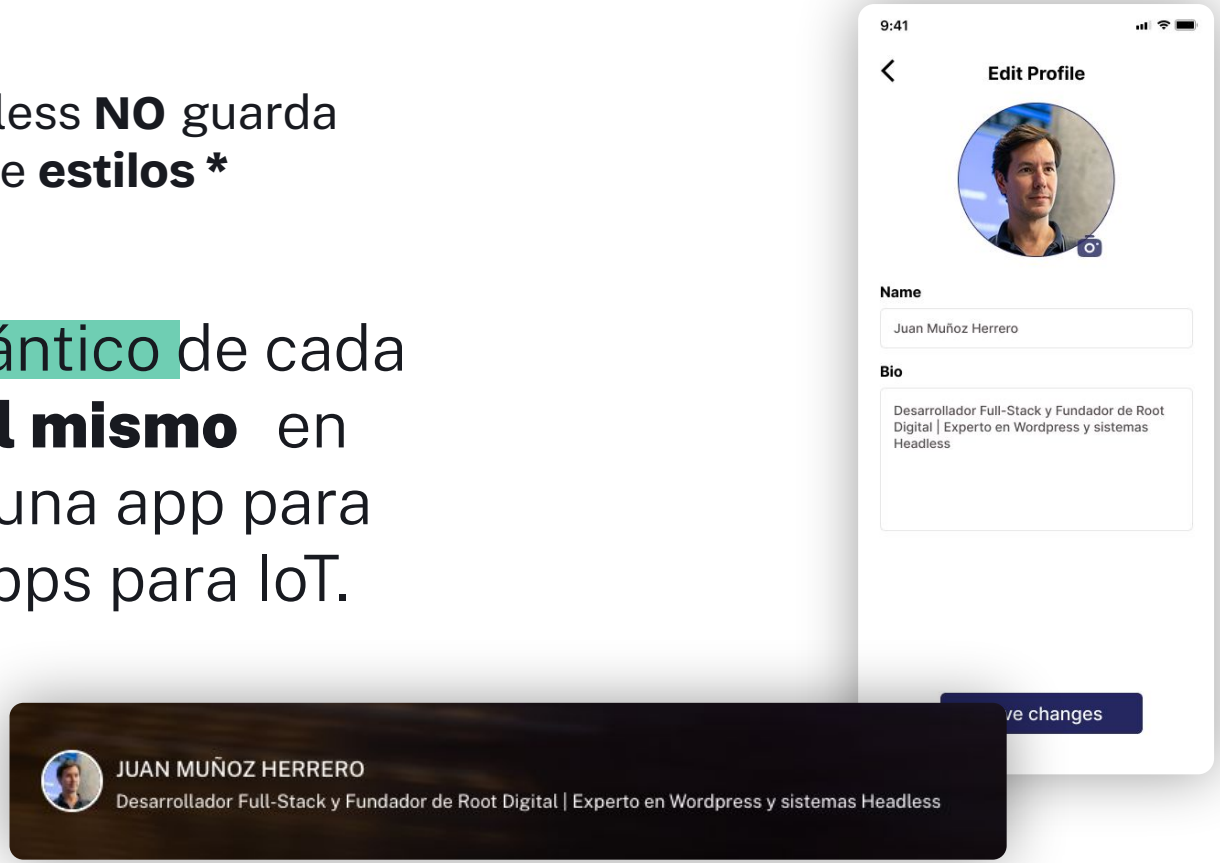


Un sistema headless guarda el contenido **en bruto**

Cada frase almacenada, tiene un **valor semántico** , más allá de su significado.

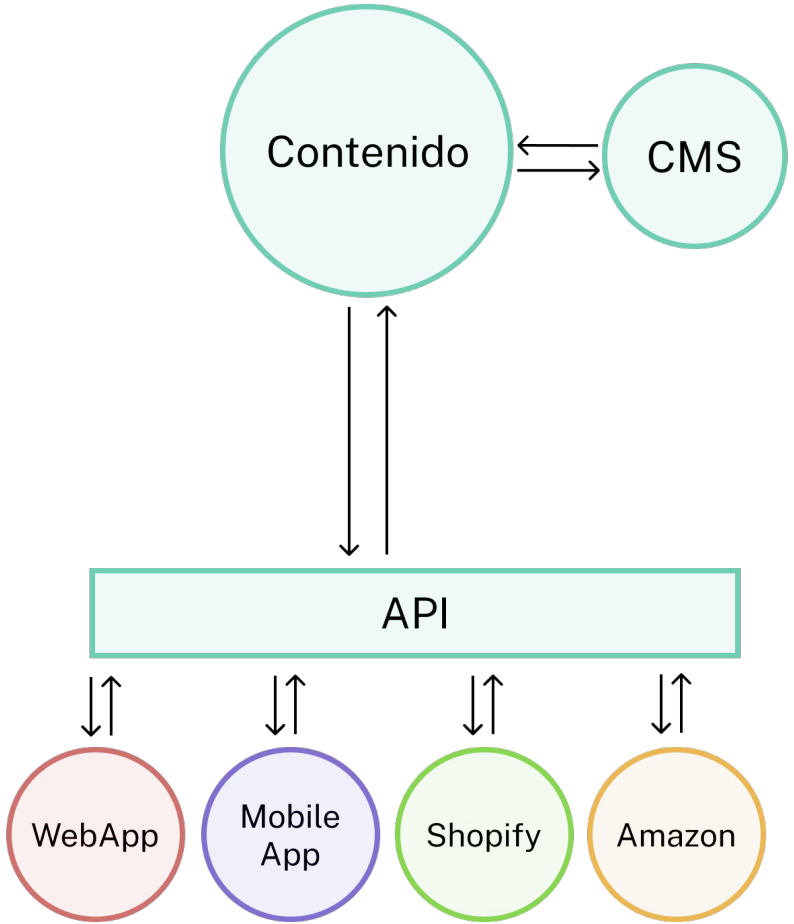
Un sistema headless **NO** guarda información sobre **estilos ***

El valor semántico de cada frase, **será el mismo** en una web, en una app para móvil, o en apps para IoT.



Nuestro CMS sería la única fuente de verdad o *source of true*

Nuestro headless CMS se convierte en un **centro de operaciones** , desde el que distribuimos el contenido hacia todas las plataformas.



Todo esto esto está muy bien, pero...

¿Cuál es la madre del cordero?

SSR - Server Side Rendering
CSR - Client Side Rendering
SSG - Server Side Generation

Server Side Rendering

Empezamos por el más común: Server Side Rendering o ***Renderizado en el Lado del Servidor***.

El HTML de la página es generado en cada petición en el servidor, y descargado al navegador del cliente. Es la forma en la que funciona una instalación estándar de WordPress.

Client Side Rendering

En este caso, la página se **renderiza en el navegador del cliente**.

El cliente se descarga una pequeña app (una sola vez), normalmente en forma de código JS, que se ejecuta en su navegador, y renderiza el HTML resultante tras la ejecución. Un ejemplo de CSR son las aplicaciones en React de grandes plataformas como Netflix, Airbnb, Spotify, etc.

Server Side Generation

Por último, ***Generación en el Lado del Servidor***.

En este caso, el servidor pre-renderiza las plantillas HTML, éstas se envían al navegador con el contenido de una página determinada, y a partir de aquí, el contenido de las siguientes páginas a las que se navega, se reemplaza en el cliente.

Es la forma en la que funcionan las páginas más modernas, hechas en **Vue.js**, **Next.js**, **Angular**, etc. Ejemplos: Toyota, Nike, El Corte Inglés, Mango...

Ventajas de los sistemas headless en el SEO

Mejor rendimiento (webs más rápidas)

Mayor control técnico (URLs, metadata, esquemas estructurados)

Facilidad para optimizar **Core Web Vitals**

Implementación de cambios en el DOM más rápidos

Reducción / eliminación de migraciones

Inconvenientes de los sistemas headless

Mayor dependencia de desarrolladores frente a soluciones populares y low-code

Mayor tiempo de integración inicial - requiere mayor planificación desde el inicio del proyecto

Posibles dificultades con algunas herramientas SEO

Casos de éxito y ejemplos reales

brightonSEO.

Casos de éxito y ejemplos reales: PUMA

1. Estructuración avanzada y reutilización de contenido

Anteriormente, el contenido estaba fragmentado en múltiples servidores locales y sistemas aislados, generando duplicación, inconsistencia y problemas para el SEO.

Con Sanity, se creó una única fuente de datos estructurada, permitiendo reutilizar más de 55,000 piezas de contenido, mejorar la consistencia, relevancia temática y facilitar el SEO a nivel global.

Casos de éxito y ejemplos reales: PUMA

2. Velocidad y flexibilidad en la implementación

Antes, implementar cambios o desplegar contenido nuevo tomaba mucho tiempo, afectando negativamente a la velocidad y agilidad necesarias para mantener buenas métricas de rendimiento.

La nueva infraestructura con Sanity y Next.js permite implementaciones rápidas (en solo 5 minutos), mejorando radicalmente el rendimiento del sitio, factor clave para SEO.

Casos de éxito y ejemplos reales: PUMA

3. Personalización de campañas y experiencia del usuario

Previamente, la personalización estaba limitada debido a la falta de integración y segmentación eficiente del contenido.

Ahora, integrando datos desde Salesforce a Sanity, es posible ofrecer **contenido más personalizado**, enfocado a diferentes audiencias y mercados, algo altamente valorado por Google debido a la mejora en la experiencia del usuario (UX).

Casos de éxito y ejemplos reales: BrightonSEO

1. Core Web Vitals y velocidad de carga

La anterior web en WordPress presentaba tiempos de carga lentos, perjudicando significativamente los **Core Web Vitals**.

La migración hacia Sanity y Next.js aportó un gran incremento en velocidad y rendimiento, factores cruciales para mejorar la experiencia del usuario y el posicionamiento orgánico.

Casos de éxito y ejemplos reales: BrightonSEO

2. Gestión avanzada del contenido

El sistema anterior limitaba la reutilización del contenido y no permitía la inclusión ágil de elementos clave para SEO (ej.: schema markup, canonicals, organización estructurada).

Se implementó un modelo estructurado de contenido que mejora la visibilidad orgánica mediante entidades como Speakers, Trainers, Talks y Courses, aumentando así la relevancia semántica y posicionamiento en Google.

Casos de éxito y ejemplos reales: BrightonSEO

3. Seguridad y estabilidad técnica

WordPress exigía la utilización excesiva de plugins, lo que generaba riesgos elevados de seguridad y **afectaba indirectamente al SEO** (malware, spam, penalizaciones).

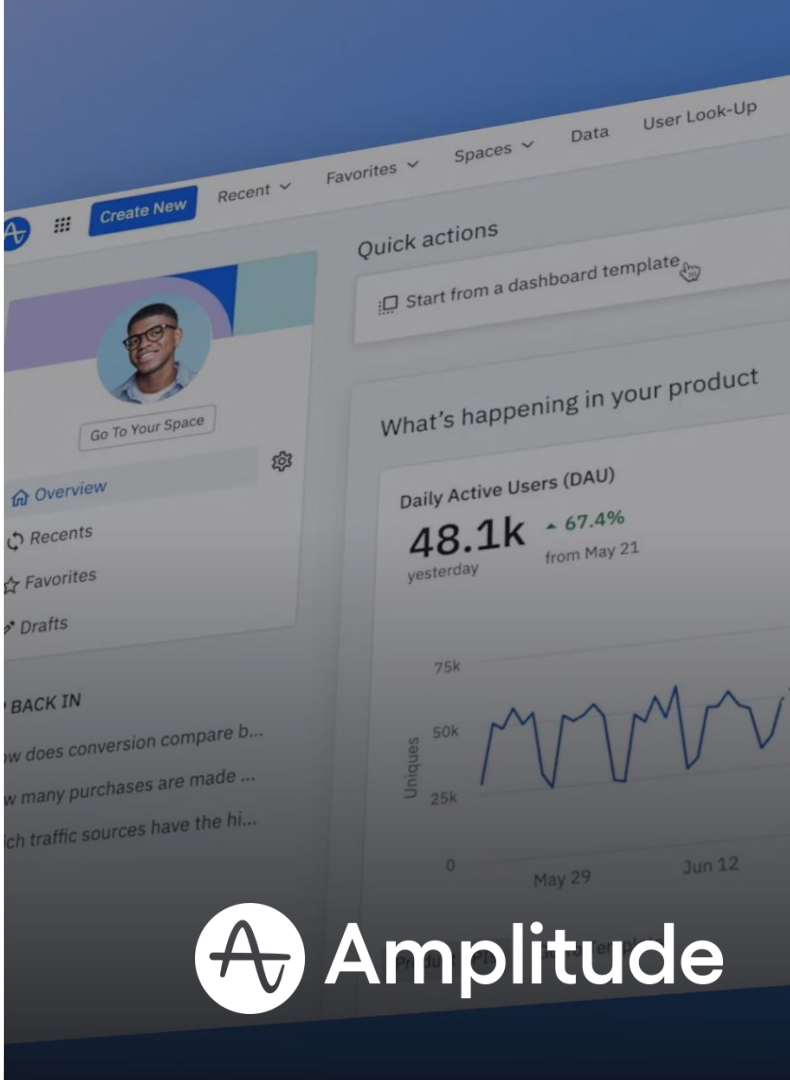
El CMS headless ofrece un entorno con menos vulnerabilidades, al reducir drásticamente la dependencia de plugins de terceros.

Casos de éxito y ejemplos reales: Amplitude

1. Incremento significativo de contenido SEO

Antes publicaban solo 4-5 piezas mensuales debido a limitaciones técnicas del CMS anterior.

Tras migrar a Sanity, aumentaron **18 veces** el contenido SEO mensual, logrando un **19% más de tráfico orgánico** interanual.

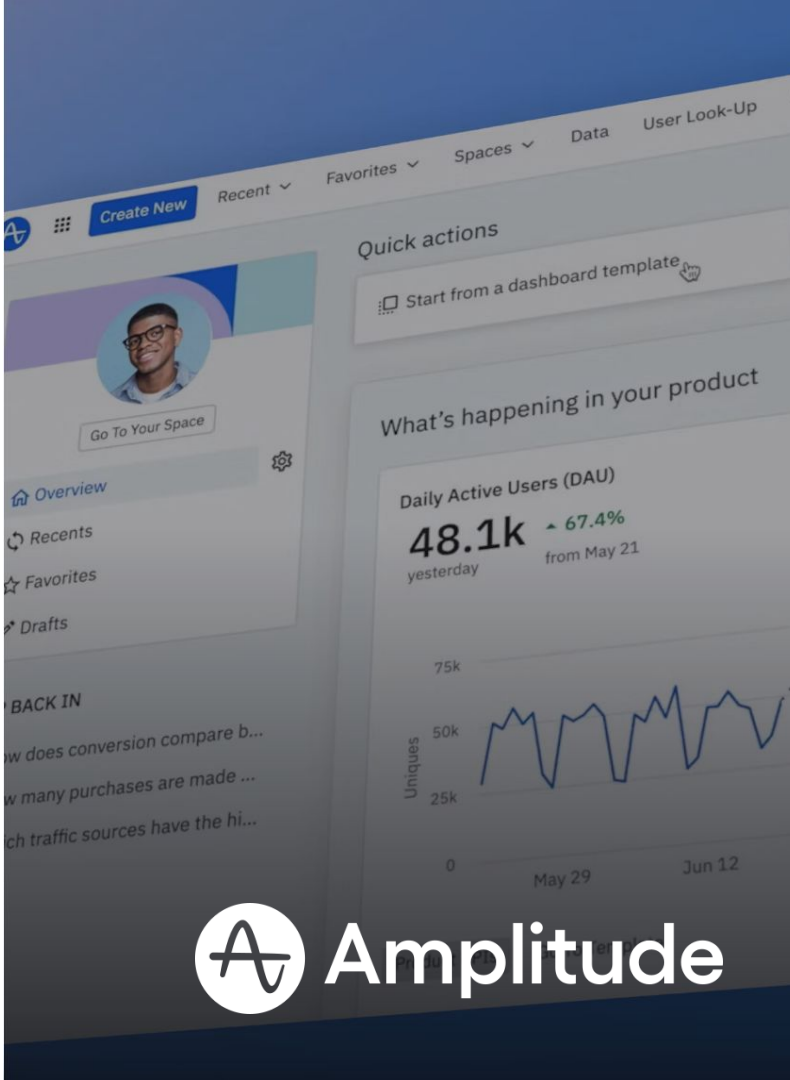


Casos de éxito y ejemplos reales: Amplitude

2. Mejora radical en velocidad del sitio

Tras migrar desde WordPress a Sanity, la velocidad del sitio mejoró en un **76%**, proporcionando una experiencia superior en términos de rendimiento y tiempos de carga.

La mejora en velocidad es fundamental para SEO, impactando directamente en las métricas de experiencia del usuario (UX) y Core Web Vitals, factores críticos en el posicionamiento.

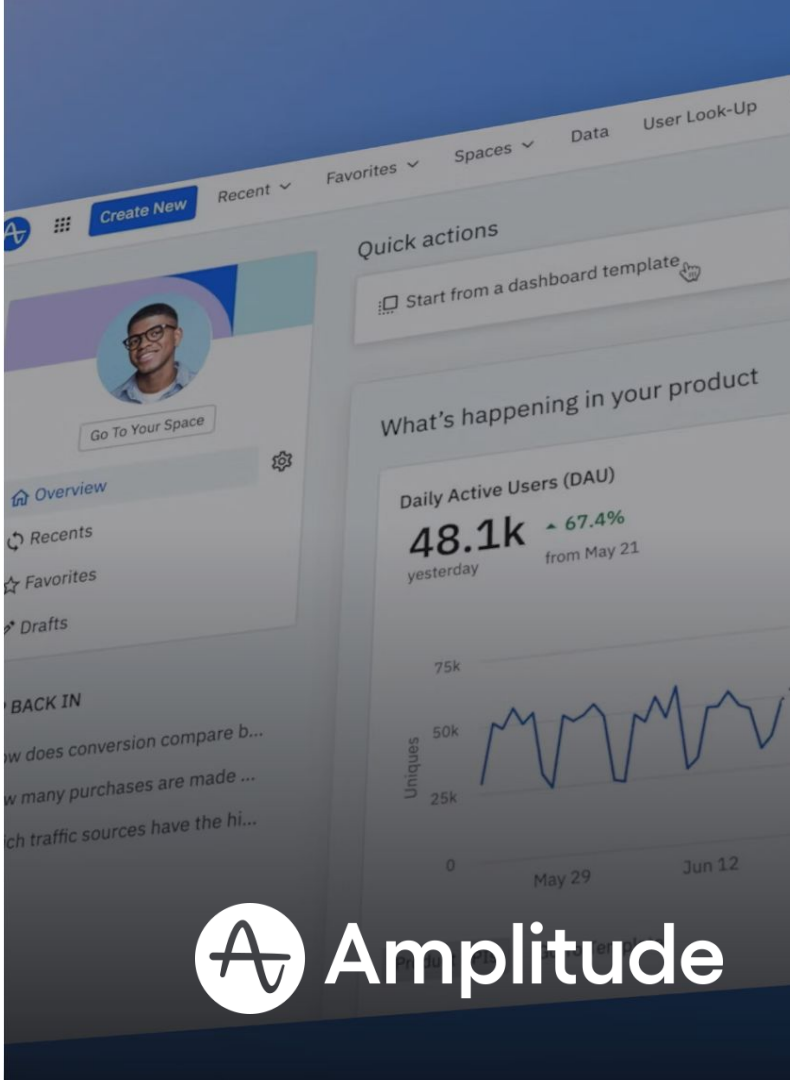


Casos de éxito y ejemplos reales: Amplitude

3. Experimentación A/B integrada y autónoma

Antes apenas podían realizar pruebas A/B por alta dependencia técnica.

Ahora los equipos hacen experimentos de forma autónoma desde Sanity, aumentando conversiones (por ejemplo: **+25%** clics en registros) y **mejorando engagement SEO**.

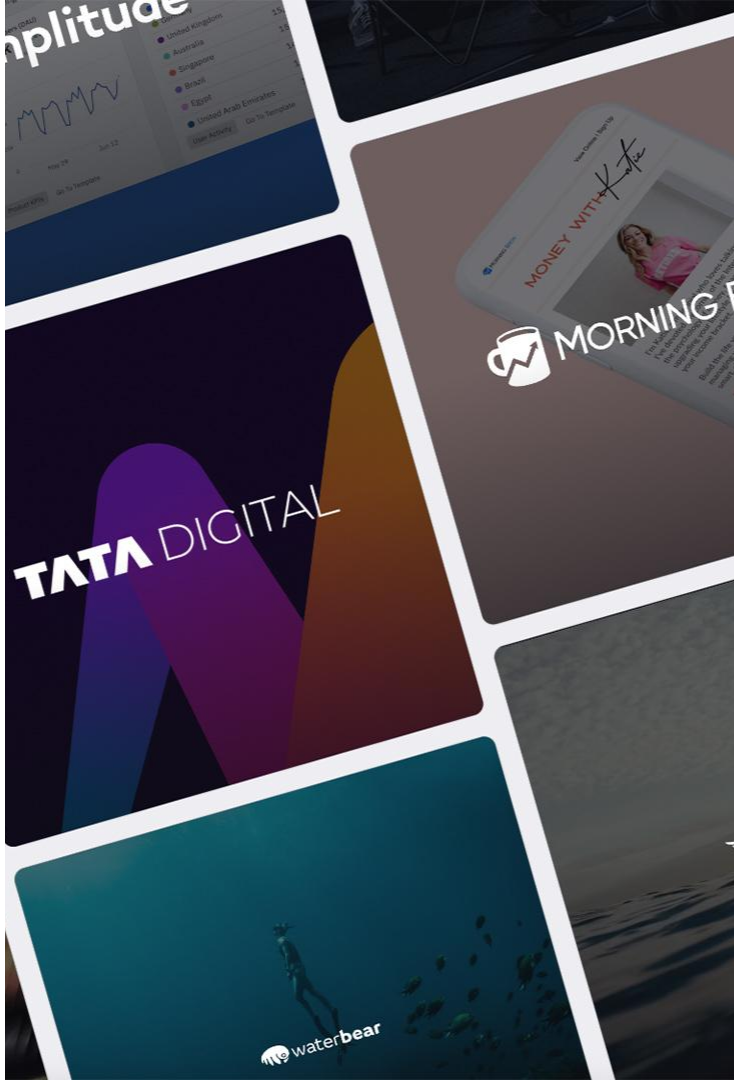


Cuándo usar un Headless CMS

Grandes sitios web, especialmente proyectos **multicanal**, y con gran volumen de datos.

Equipos técnicos y de SEO independientes, pero bien coordinados.

Recursos necesarios, tiempo, inversión, perfil técnico, requisitos de seguridad...



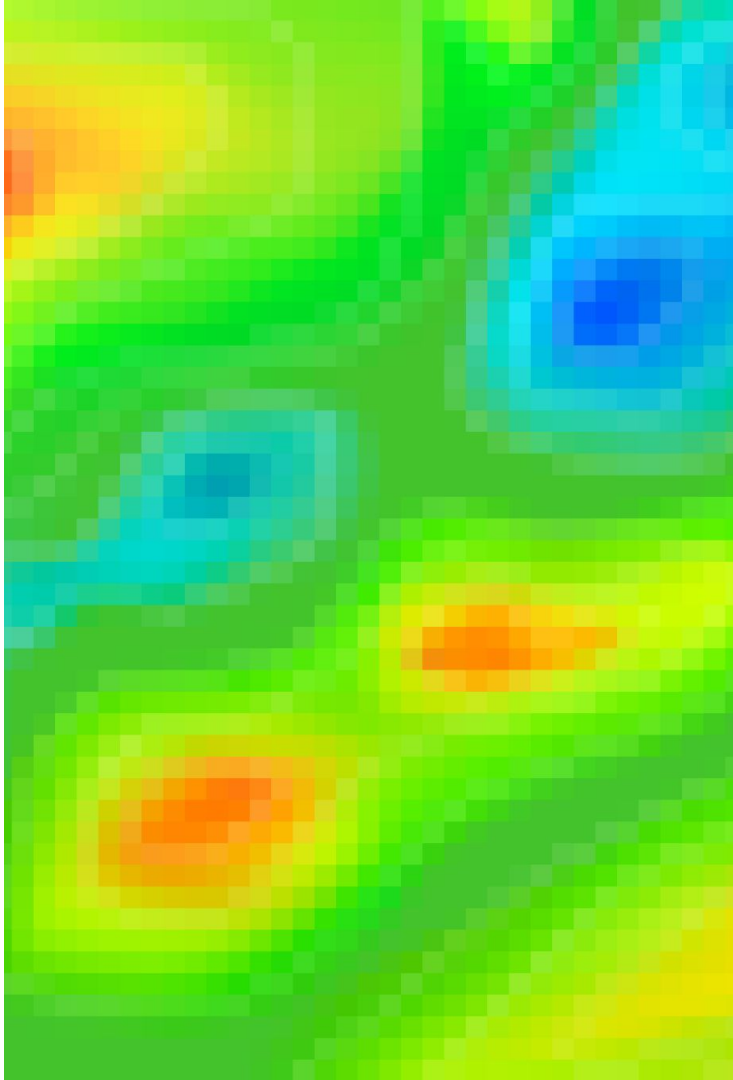
Conclusiones y recomendaciones finales

Beneficios demostrados indirectamente

Mejora significativa en velocidad:

Google afirma oficialmente que **la velocidad de carga y la experiencia del usuario son factores de ranking**.

Los sistemas headless facilitan la optimización técnica del frontend y, por ende, mejores resultados SEO.

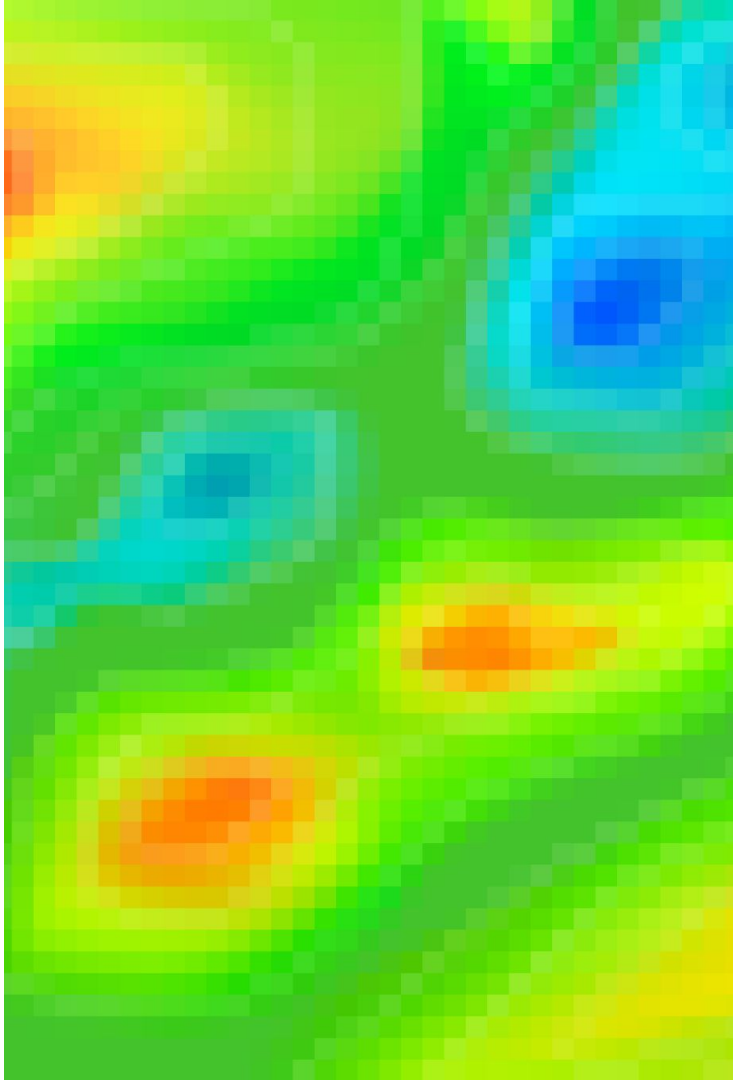


Conclusiones y recomendaciones finales

Beneficios demostrados indirectamente

Flexibilidad técnica para SEO:

Se pueden controlar al detalle elementos clave para SEO como etiquetas meta, schema markup, URLs personalizadas y datos estructurados, gracias al control absoluto sobre el frontend.



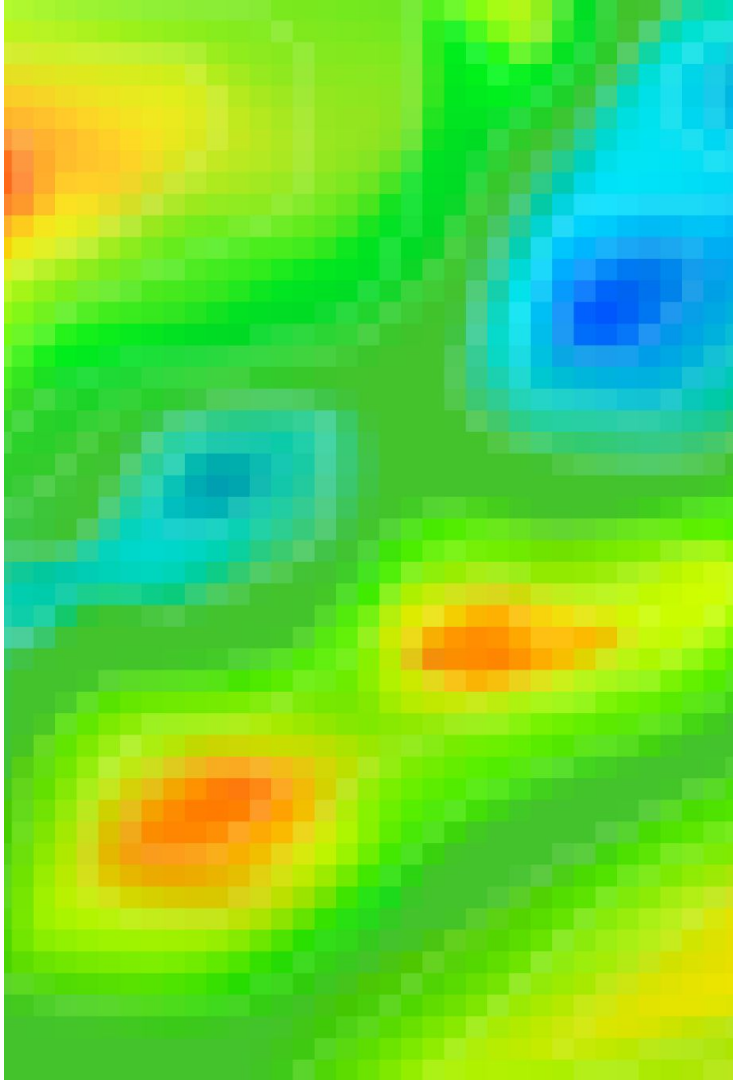
Conclusiones y recomendaciones finales

Beneficios demostrados indirectamente

Indexación más eficiente:

Google recomienda claramente sitios con contenido accesible y bien estructurado.

Con un Headless CMS es más fácil aplicar prácticas de código limpio y evitar bloqueos involuntarios en la indexación.

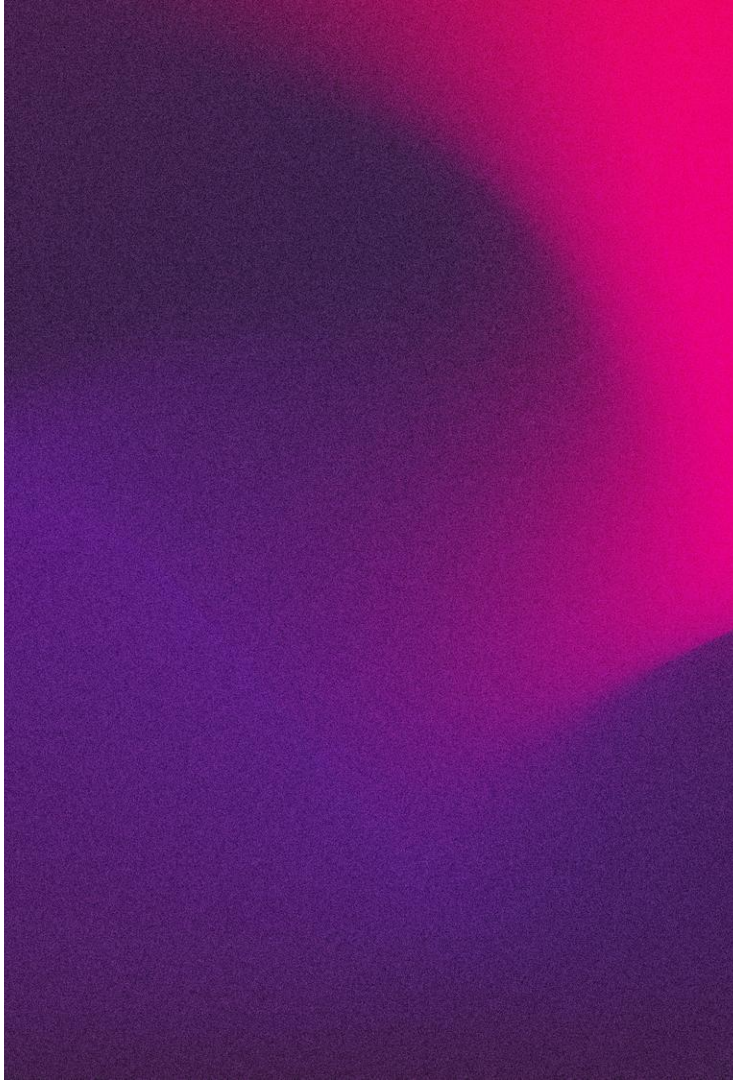


Conclusiones y recomendaciones finales

Consideraciones importantes

La mejora SEO al adoptar un CMS Headless NO es automática:

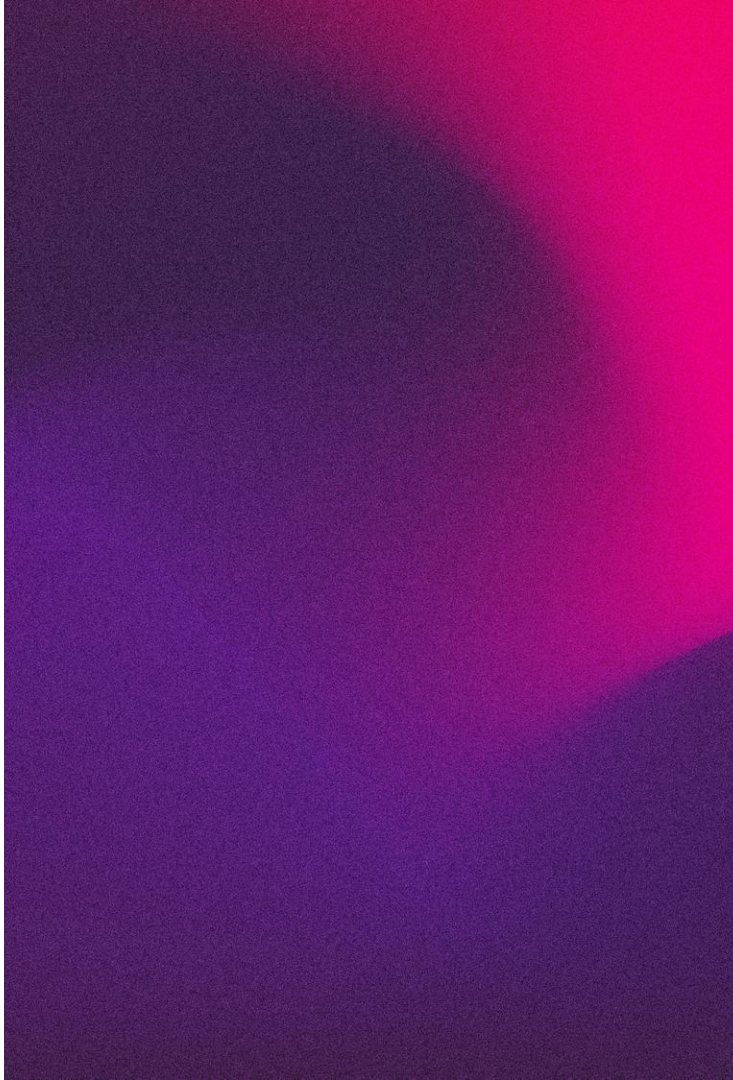
Depende mucho del equipo técnico y de cómo se implemente el frontend, el CDN sobre el que se despliegue, servicios externos, etc.



Conclusiones y recomendaciones finales

Consideraciones importantes

No se trata solo del CMS, sino de la **tecnología frontend y arquitectura** elegida (Next.js, Gatsby, React, Vue, etc.).

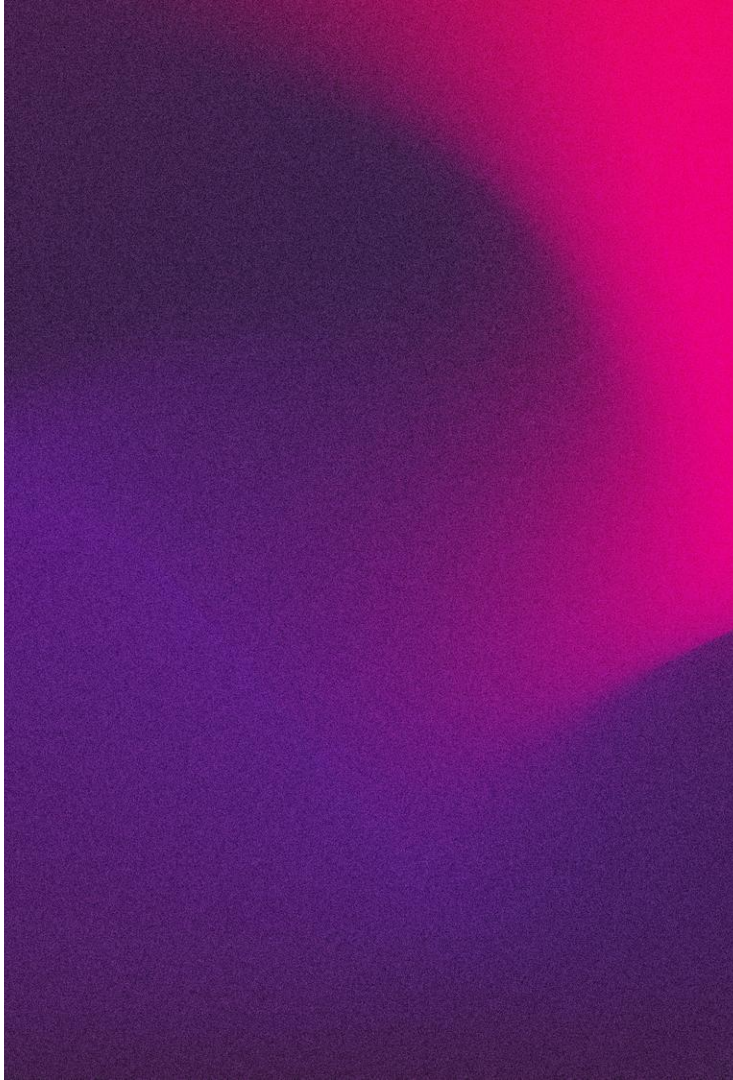


Conclusiones y recomendaciones finales

Consideraciones importantes

Google oficialmente **nunca** recomienda una tecnología específica, sino buenas prácticas técnicas.

La mejor tecnología o el mejor sistema, es el aquel con el que te sientas cómodo, te permita resolver un reto en el menor tiempo posible, y no genere muchos problemas.



¡Gracias!

headless-agency

juan@rootdigital.agency

Juan Muñoz

Fundador de Root