



Coders Cantabria

SEO para SPA

Sí, puedes usar Javascript en tu web y no romper el SEO.

Alfonso "Mou" Moure

¿Qué vamos a ver hoy?

1. Render, Chrome y Googlebot

Qué es eso de una SPA y porqué tiene particularidades que afectan a SEO. Y, ya de paso, qué es eso del SEO.

2. SEO dentro de la SPA

La SPA debe saber cómo hablar a Google. Y, casi más importante, debe saber hablar rápido y sin atascarse (demasiado).

3. Liadas gordas, mitos y leyendas

El render se olvida o pierde cosas por el camino, cookies indigestas y las versiones dinámicas separadas creadas

por Cthulhu.

4. Herramientas y trucos

Algunas conocidas, otras menos habituales: vamos a nutrir nuestra caja de herramientas con nuevos *goodies*.

1. Render, Chrome y Googlebot

¿Qué es eso del SEO y a mí qué me importa?

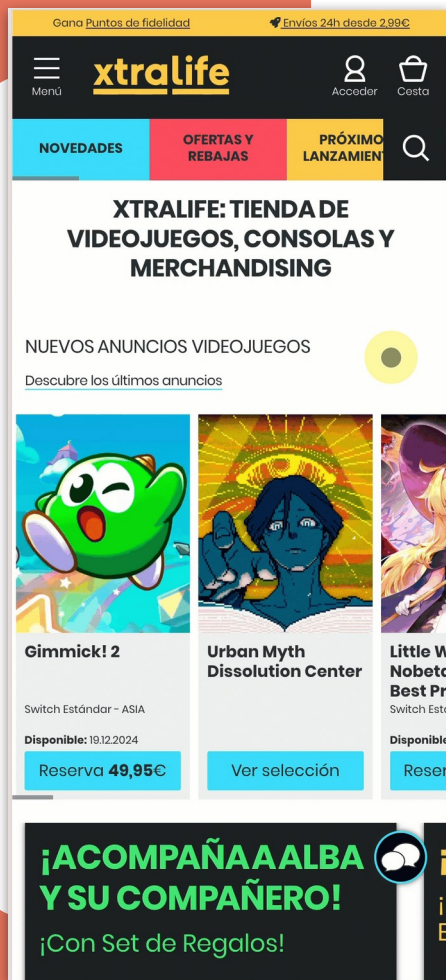
Los SEOs ~~son~~ somos muy pesados. Pero lo
somos por algo.

SEO

Optimizar cómo nos ve un buscador a la hora de indexar el contenido de nuestra web y “molarle” para aparecer más arriba.

¿Por qué más arriba?

A más arriba, más clics. Más clics, más tráfico. Más tráfico, más capacidad de generar negocio. A más negocio, más dinero. A más dinero, más contentos .

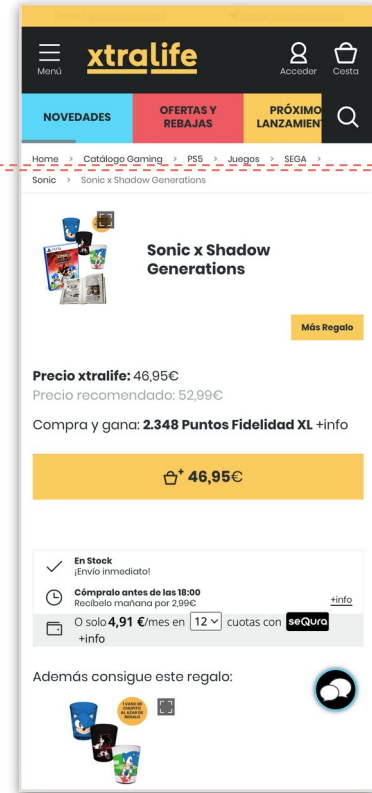
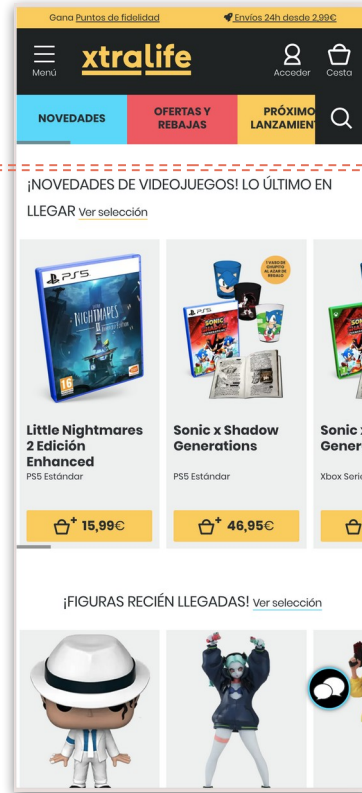
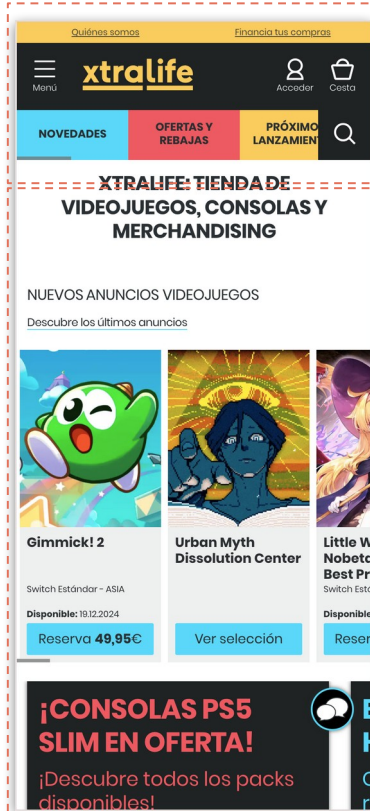


¿Qué es eso de una SPA y qué tiene que ver con el SEO?

SPA = Single Page Application

Carga inicial de página seguida de una carga parcial por bloques. Menor tráfico de red y mejora de UX.

Estático



Dinámico

View 1

View 2

View 3

Google y Googlebot

Google utiliza un bot conocido como Googlebot que descarga URLs, recopila sus contenidos y los procesa.

¿Puede una máquina usar la SPA?

Claro, pero ¿cuánto cuesta interactuar con ella? ¿Es rentable? ¿Cuánto hay que esperar para saber si todo está cargado? ¿Puede Google esperar?

Mola. ¿Qué tiene que ver todo esto con el SEO?

¿Cómo analiza Google el contenido de una página?

Headless browser basado en Chrome

Googlebot se basa en Chrome Stable desde 2015. Es actualizado vía integración continua con cada nueva versión.

Render completo de página y diferentes viewports

El headless carga la página al completo y ejecuta el render de manera normal para visualizarlo como un usuario real.

Alto coste de render y análisis

Renderizar una página lleva tiempo. ¿Imaginas tener que hacer esto miles de millones de veces al día?

¿Puede el navegador del bot ejecutar el render completo?

Si Googlebot intenta ver la web como un usuario, es importante que le brindemos la capacidad de hacerlo.

¿Puede hacerlo lo bastante rápido como para ser rentable?

Como el render cuesta dinero, es obvio que Google no desperdiciará ni recursos, ni usuarios, en una web lenta.

¿Puede descargar el contenido?

Para poder renderizar el contenido, el bot debe poder descargar todos los recursos esenciales para ello.

¿Qué efecto tiene una
SPA sobre cómo
Google ve la web?

2. SEO dentro de la SPA

¿Qué debo hacer para
que el bot pueda usar mi
SPA?

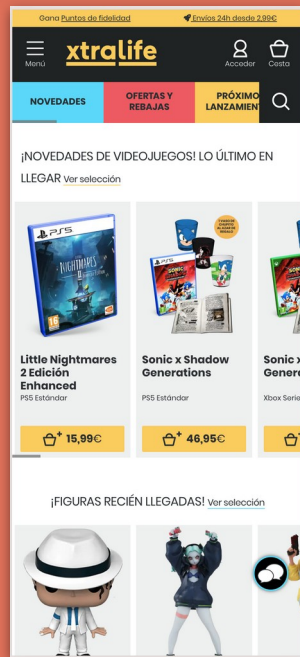
Render, contenido y rastreabilidad

Primero: ¿cómo
llega el bot a mi
página y cómo salta
de un sitio a otro?

<https://www.xtralife.com/novedades>

URL única para acceder a cada
view

Como es un bot *stateless*, cada view de tu
SPA debe ser accesible mediante una URL
completa.



Primero: ¿cómo
llega el bot a mi
página y cómo salta
de un sitio a otro?

Disponibilidad de enlaces en el front-end

Todos los cambios de view deben hacerse mediante un enlace HTML con su precioso atributo *href*.

```
<a  
  href="/producto/.../83395"  
  [...]  
>  
  ...  
</a>
```

Primero: ¿cómo llega el bot a página y cómo de un sitio a otro

¿No puede el bot usar enlaces
basados en JS sin un A?

Solo aquellos que, de manera muy
evidente, hacen uso de una URL de
destino para su ejecución.

Con sinceridad 🐱

A mí no me gusta nada usar este tipo de
soluciones. Mejor un atributo *href* con
una etiqueta A: esto no me ha fallado
jamás. Con Google siempre es mejor
KISS.

```
en('/producto/.../83
```

State

```
<div  
  onclick="history.pushState(null  
    , '', '...'); (...)">  
  ...  
</div>
```

Primero: ¿cómo
llega el bot a mi
página y cómo salta
de un sitio a otro?

Mucho ojo con la navegación
basada en cookies

El headless browser es *stateless*: no
recuerda lo que ha visto antes. No hagas
que la navegación dependa de cookies.



Segundo: ¿cómo sabe el bot si está en el lugar correcto?

Googlebot entiende de códigos de estado

Devuelve el código de estado correcto o redirige a otro lugar donde pueda devolverse. Ojo con los errores en 200. Hay que forzar que sea el servidor quien responda el error.

200

<https://www.xtralife.com/me-lo-invento>

Location change

404

<https://www.xtralife.com/error>

```
if (window.location.pathname === '/me-lo-invento') // y otra lógica
{
    // Redirige a la URL de error, cargando el nuevo recurso desde el servidor
    window.location.replace('/error');
}
```

Segundo: ¿cómo sabe el bot si en el lugar co

⚠ Cuidado con el *tragabolas*

Nunca redirijáis con esta solución a una URL que no tengáis seguro que existe porque puede suponer un riesgo de seguridad y para SEO.

Uso de redirecciones desde Javascript enviadas por la SPA

Cuando una URL cambia es preciso que Google lo sepa para evitar que pida la vieja y devuelva un 404. Para ello, usamos *location change* para enviar a my que redirige "a donde

<https://www.xtralife.com/url-vieja>

Location change

[https://www.xtralife.com/redir?url=\[url_nueva\]](https://www.xtralife.com/redir?url=[url_nueva])

301

<https://www.xtralife.com/url-vieja>

200

Redirección

Tercero: ¿puede Google verlo to



Cuidadito con el robots.txt sobre recursos del front-end...

Dado que Googlebot necesita renderizar el contenido y verlo perfecto, no debemos bloquear los recursos clave.

Ficheros CSS y Javascript

Son fundamentales para el render y, por lo tanto, debemos vigilar que nunca estén bloqueados en ningún nivel de robots.txt

Imágenes

Mucho ojo: las imágenes son importantes. Tampoco deben estar bloqueadas de ninguna manera.

Tercero: ¿puede Google verlo todo?

¡Y más cuidado todavía con
recursos del back-end!

Las SPA usan llamadas al back-end para
cargar contenidos parciales

APIs internas: ¡nada de robots!

Si bloqueas las llamadas a las API
internas, corres el riesgo de dejar el front,
vacío.

Cuarto: ¿qué tal andamos de contenido? ¿Y de HTML?

¿Debe interactuar el usuario para ver contenido?

¿Tienes desplegables? ¿Quiéres que Google indexe su contenido? Carga su texto y recursos desde el principio.

Modales

Caso particular, pues no siempre se usan. Aplica la misma norma: ¿son interesantes para SEO? No esperes al usuario.

Imágenes

El *lazy-loading* es válido. Lo que no lo es es crear su etiqueta con interacciones del usuario.

Cuarto: ¿qué tal andamos de contenido? ¿Y de HTML?

¿Y los recursos externos? Mapas, vídeos, etc.

Ese tipo de contenido no se indexa como tal.

TL;DR: Mi consejo suele ser cargar un preview en formato imagen con el mismo tamaño para que el render esté completo y activar el recurso dinámico con una interacción del usuario. Esto permite evitar el front-end tenga que cambiar de tamaño (con efecto negativo en web vitals: otro tema interesante), mientras que el usuario sigue pudiendo obtener su experiencia completa.

Cuarto: ¿qué tal andamos de contenido? ¿Y de HTML?

Carga de contenidos clave, con o sin Javascript

TITLE, Meta Description, canonicals, meta robots tag, datos estructurados de Schema y otros: con o sin Javascript, pero cárgalos.

Tanto al inicio, como en cada view

Carga su contenido siempre y actualiza su contenido cuando haga falta mediante Javascript sin ningún miedo.

Cinco: vísteme despacio, que tengo prisa

La SPA inicial pesa. Pesa mucho.
¿Cuánto esperará Google?

Si tu contenido no está disponible al 100% en menos de 5 segundos deberías activar todas las alarmas.

Ojo con web vitals: CLS

Siempre son importantes, pero vigila el CLS de tu SPA: es mucho más vulnerable si tu front-end no está bien ajustado.

¿Qué tal tu LCP y FID?

Pre-reserva el espacio de tu *largest contentful paint* y monitoriza tu *first input delay* (FID) para asegurar web vitals y la calidad de la experiencia de usuario.

3. Liadas gordas, mitos y leyendas

Algunas liadas pardas adicionales que he visto en mi trabajo

O como jorobar un trabajo bien hecho con toda la buena intención.

“Separemos mobile y desktop para facilitar adaptarlos”

Amigas, amigos, para esto está el responsive. Si no tienes una buena razón, no lo hagas.

¿Por qué digo esto con una SPA?

Imaginad un caso en el que las redirecciones funcionan bien en la SPA de desktop pero no en mobile. ¿Qué hará Google? Nada bueno.

Catástrofe 1: separar la web mobile de la de desktop y olvidarte del asunto

Catástrofe 2: usar SSR (Server-Side Render) y pensar “esto ya está apañao”

SSR: bonito, queda genial. Caro.

Googlebot ejecuta Javascript a la perfección. ¿Por qué invertir en SSR? Si te lo puedes permitir, está estupendo.

“La agencia SEO dice que necesitamos SSR”

Todo lo expuesto como reglas SEO para SPA sigue siendo válido si usas una SSR. No te confíes: aplica todo lo necesario para trabajar y no te confíes con tus €€€.

“Ponemos un listener para cuando acabe y listo”

Mucho ojo con usar *requestIdleCallback*: el Chrome de Googlebot no tiene periodos idle y nunca sucederá. ¡No para nunca!

“Ponemos un listener de scroll y apañao”

Si quieres apañar que algo fundamental no cargue parece una buena idea. Si prefieres que funcione, recuerda: el contenido clave no debe esperar una interacción.

Catástrofe 3: el listener de la muerte al que nadie escucha

Catástrofe 4: no hacer uso de try-catch y dejarlo todo a la buena de voluntad del navegador (nota: no la tiene)

“Solo es un request, no pasa nada”

¿Y si la llamada a la API falla? ¿Qué sucede si no hay respuesta o tarda demasiado? No bloquees el navegador.

“Ya miraremos lo de la licencia de Google Maps”

¿Qué pasa si la licencia de un bloque ha caducado? ¿Y si hay dependencias que disparan excepciones no controladas? 🙄

Código repartido entre varios
ficheros: útil, ordenado, malo

Ojo, malo para SEO. Me gusta la
abstracción, pero a Google no le gusta
descargar de más. Ojo con el *crawl rate*.

¿Es el último o más pesado CSS o JS el
más importante?

Es posible que nunca sea descargado. Si
ves Google Search Console cortado, es
que algo puede pintar mal en este
sentido.

Catástrofe 5: divide, vencerás y la liarás

Catástrofe 6: el aviso de cookies que lo mató todo

Aviso de cookies que bloquea algo más que cookies

¿Necesita tu web consent? Claro.

¿Bloquea la carga de algo no tenerlo?

Peligro. Evítalo.

¿Ocupar toda la pantalla es positivo? Sí para el consent, no para SEO

Usa Google consent mode en tu aviso de cookies para que el bot pueda adaptarse a tus necesidades. ¡Y no desplaces componentes!

4. Herramientas y truques

¿Qué sería una charla
SEO sin alguna
recomendación de
herramientas?

A la gente de SEO no le gusta demasiado programar (¡a mí sí!), pero tienen herramientas fantásticas para mantener su excusa.

https://www.xtralife.com/producto/the-legend-of-zelda-echoes-of-wisdom-switch-estandar/90410

URL Inspection

GOOGLE INDEX LIVE TEST

Tested on: 26 Oct 2024, 07:52

✓ **URL is available to Google**
If it gets indexed and selected as canonical, it could appear in Google Search results with all relevant enhancements. [Learn more](#)

VIEW TESTED PAGE Page changed? REQUEST INDEXING

✓ **Page availability** Page can be indexed

Enhancements and experience

✓ Product snippets	2 valid items detected Non-critical issues detected	>
✓ Merchant listings	2 valid items detected Non-critical issues detected	>
✓ Breadcrumbs	1 valid item detected	>
✓ Review snippets	2 valid items detected	>
✓ Sitelinks searchbox	2 valid items detected	>

Tested page

HTML SCREENSHOT MORE INFO


¡Gana Puntos de Fidelidad! [Gana 3,4h desde 2.99€](#)

Menu **xtralife** Acceder Cesta

NOVEDADES OFERTAS Y REBAJAS PRÓXIMO LANZAMIENTOS

Home > Catálogo Gaming > Switch > Juegos > Nintendo >


The Legend of Zelda > The Legend of Zelda: Echoes of Wisdom




The Legend of Zelda: Echoes of Wisdom

Precio xtralife: 54,95€
Precio recomendado: 59,99€

Compra y gana **2.748 Puntos Fidelidad XL** +info

 **54,95€**

En Stock  ¡Envío inmediato! +info

En Stock
¡Envío inmediato!

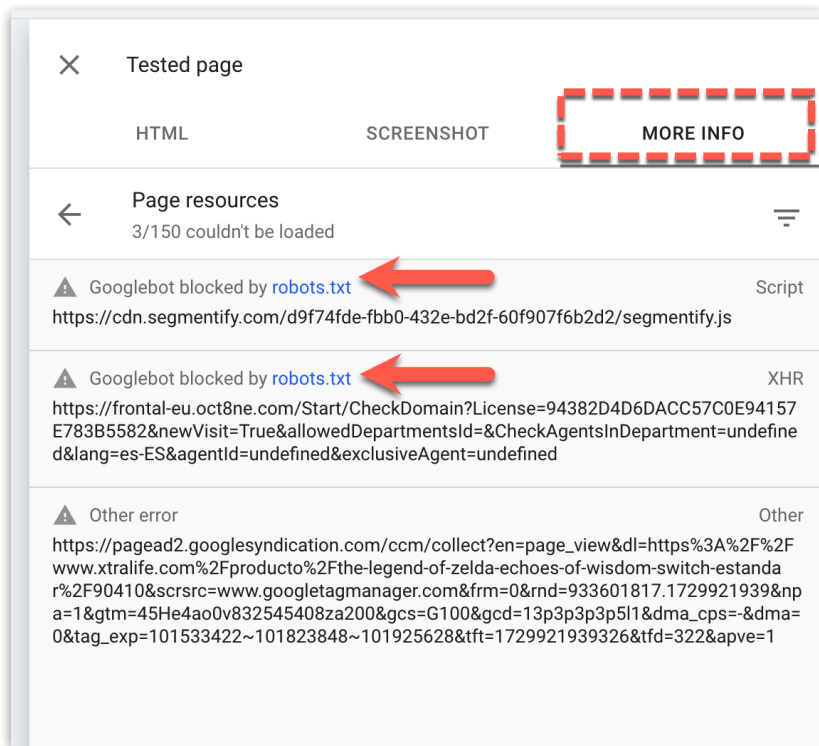
Cómpralo antes de las 18:00
Recíbelo mañana por 2,99€

O solo **5,57 €/mes** en 12 cuotas con **seQuira** +info

Rendered with Google Inspection Tool smartphone

Google Search Console

<https://search.google.com/search-console/>



Google Search Console

<https://search.google.com/search-console/>

Chrome “Canary” DevTools

<https://www.google.com/chrome/canary/>



Desactivar Javascript para pruebas
Bloquear contenidos en la descarga
Probar con diferentes user-agents
Ver lo que está por venir en Chrome

Choose:

Console Output: [📄 Copy](#)

[illegible]

<https://tamethebots.com/tools/fetch-render>

Fetch & Render

The screenshot displays the Dentsu Fetch & Render tool interface. The left sidebar contains navigation options: Crawling, Rendering, Mobile SEO, International SEO, Local SEO, SERP & Ranking, Schema Generator, Resources, Docs, Blog, Podcast, and Presentations. The main panel is titled 'Fetch & Render' and shows the following configuration:

- URL: <https://www.xtralife.com/novedades>
- User Agent: Googlebot Smart...
- Obey robots.txt: ☒
- Render: ☒
- Proxy Load Timeout (s): 5
- Headless Browser: Chromium 79
- Accept Language: Spanish
- Region: Spain
- Username: [redacted]
- Password: [redacted]

A 'FETCH & RENDER' button is located at the bottom right of the configuration section.


Below the configuration, a table shows the results of the fetch and render process:

URL	Crawlable	Status	Redirect Type	Indexable
https://www.xtralife.com/novedades	✓	200 OK		✓

The bottom section displays the 'HTTP RESPONSE - 0.50 SEC' and 'RENDERED PAGE - 2.65 SEC'. It shows the raw HTML code on the left and a preview of the rendered page on the right. The HTML code includes Google Tag Manager scripts, a dataLayer, and meta tags for charset, title, and viewport. The rendered page preview shows the xtralife website header and a section titled 'NOVEDADES DE VIDEOJUEGOS LO ÚLTIMO EN LEGAR'.

<https://technicalseo.com/tools/fetch-render/>


Fetch & Render

 **Pre-rendering Testing Tool**

HTML snapshots, pre-rendering, [dynamic rendering](#)... regardless the name and specifics of the implementation, the result is usually a different HTML response based on the user agent making the request. Use this tool to check what content is being served to different user agents.



URL

User Agents

Googlebot Smartphone, Googlebot, PageSpeed I... 

TEST

Search...

User Agent	Status	Redirect	Page Title	Meta Description
Googlebot Smartphone	200 OK		Novedades xtralife	xtralife Ponemos a tu disposición una tienda de videojuegos al completi
Googlebot	200 OK		Novedades xtralife	xtralife Ponemos a tu disposición una tienda de videojuegos al completi
PageSpeed Insights (Mobile)	200 OK		Novedades xtralife	xtralife Ponemos a tu disposición una tienda de videojuegos al completi
PageSpeed Insights (Desktop)	200 OK		Novedades xtralife	xtralife Ponemos a tu disposición una tienda de videojuegos al completi
Chrome (Android Mobile)	200 OK		Novedades xtralife	xtralife Ponemos a tu disposición una tienda de videojuegos al completi
Chrome (Windows Desktop)	200 OK		Novedades xtralife	xtralife Ponemos a tu disposición una tienda de videojuegos al completi

<https://technicalseo.com/tools/fetch-render/>

Alfonso
Moure



Alfonso Moure

Muchas gracias

Añádeme a LinkedIn

¿Nos conectamos? Aquí tienes mi perfil:
<https://www.linkedin.com/in/alfonsomoure/>

Alfonso Moure Ortega

Technical SEO & WPO & Data
Engineering | Python is my third l...

