

Nuxt.js en el desarrollo web

ADI 2022-23



Víctor Hernández Justamante

Alejandro Company Rincón

Robert Kazaryan

ÍNDICE

1. ¿Qué es Nuxt.js?	3
2. ¿Para qué se utiliza Nuxt.js?	3
3. ¿Cómo funciona Nuxt.js?	3
4. Beneficios de Nuxt.js	6
5. Cómo crear una aplicación Nuxt.js	7
6. Reutilización y Atomic Web Design	8
7. Bibliografía	10

1. ¿Qué es Nuxt.js?

Nuxt.js es una librería JavaScript gratuita y de código abierto basada en Vue.js, Node.js, Webpack y Babel.js. Nuxt está inspirado en Next.js, que es un framework con un propósito similar, basado en React.js. Se anuncia a sí mismo como un “Meta-Framework para aplicaciones universales”.

Está enfocado a ayudar a desarrolladores con un entorno rápido, fácil y de manera organizada.

Su salida inicial fue el 26 de Octubre de 2016, por Alexandre Chopin, Sebastien Chopin y Pooya Parsa. Su última versión estable, 3.0.0, salió el 16 de Noviembre de 2022.

2. ¿Para qué se utiliza Nuxt.js?

La meta del framework es permitir a los usuarios crear vistas web en JavaScript utilizando el single file component de Vue.js y que pueda funcionar en ambos casos: Primero vistas de SPA que ejecuta en navegador y segundo vistas webs renderizadas en servidor que son (después de renderizar el servidor) “rehidratados” con todo el potencial SPA.

Nuxt.js ofrece la capacidad de crear una amplia variedad de aplicaciones. En su página web, puedes encontrar numerosos ejemplos de aplicaciones creadas con Nuxt.js (<https://nuxt.com/showcase>).

3. ¿Cómo funciona Nuxt.js?

Nuxt.js funciona de la misma manera que un framework del lado del servidor cuando un usuario visita un sitio web. Si el renderizado del lado del servidor está habilitado, las solicitudes se procesan en el servidor cada vez que el usuario solicita una página, por lo tanto, se necesita un servidor para poder servir la página en cada

solicitud. Además, también podemos habilitar el renderizado del lado del cliente, así, el contenido de la página se generará en el navegador usando JavaScript.

Podemos destacar tres principales acciones y métodos utilizados en Nuxt.js:

- *nuxtServerInit* (Action): es la primera etapa del ciclo de vida que se llama en el lado del servidor si el Vuex 'store' está habilitado. Es una acción de Vuex que se llama solo en el lado del servidor para popular previamente el 'store' y, finalmente, se puede usar para enviar otras acciones en el Vuex 'store'.
- *validate()* (Function): el método de validación se llama antes de renderizar los componentes de la página. Es útil para validar parámetros dinámicos de un componente de página.
- El método *asyncData()* se usa para obtener datos y representarlos en el lado del servidor, mientras que el método *fetch()* se usa para llenar el 'store' antes de mostrar la página.

Por ejemplo, cuando un sitio Nuxt.js recibe una petición inicial a la página, llama a la acción *nuxtServerInit* para actualizar los estados de Vuex (si está definido en el 'store' de Vuex). De lo contrario, si no está definido el *nuxtServerInit*, pasa a una etapa diferente.

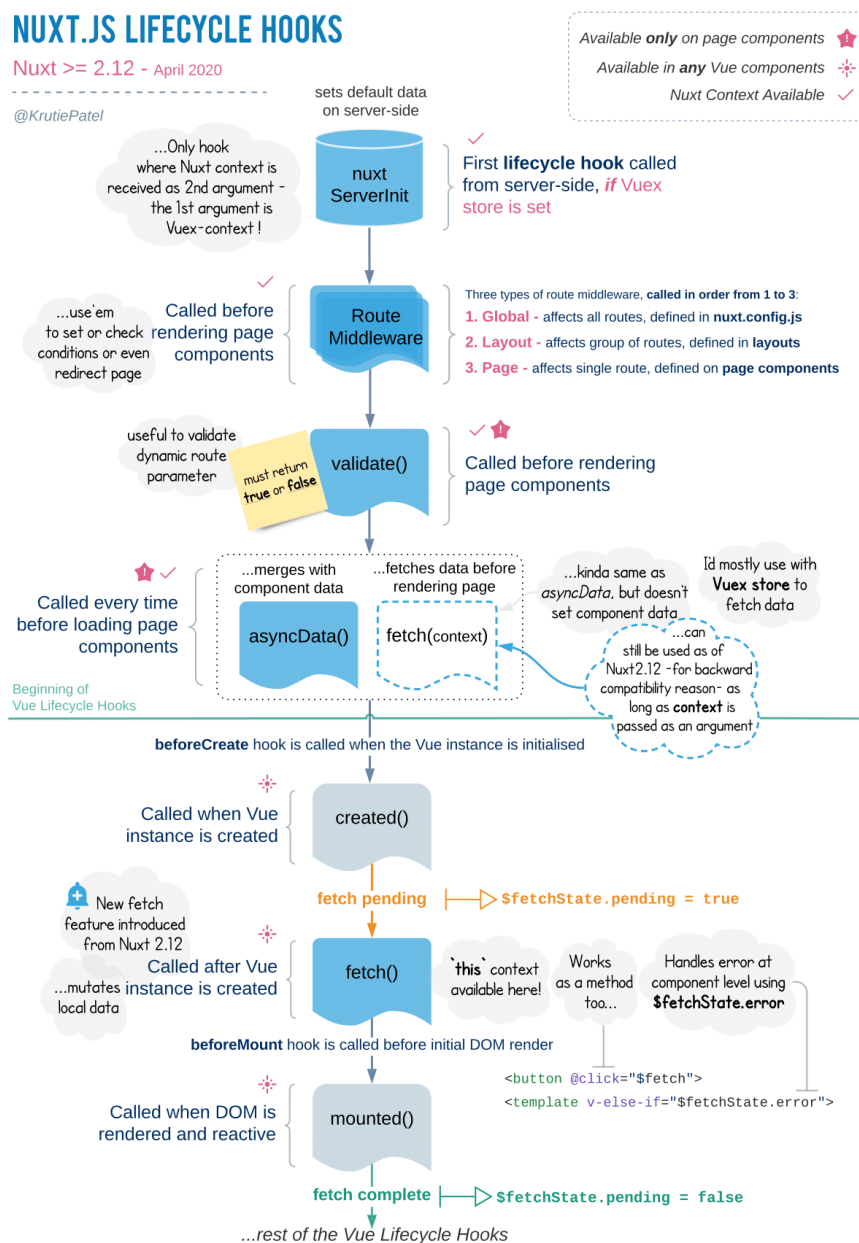
A continuación, se busca middlewares en el siguiente orden. Primero, verifica el archivo *nuxt.config.js* en busca de cualquier middleware global. Si está definido, se ejecutará antes de comprobar las páginas de *layout* en busca de middleware. Por último, ejecutará el middleware individual de cada página, incluidos sus descendientes.

Después de ejecutar todos los middlewares en orden, verificará si la ruta es dinámica y ejecutará el método *validate()* creado y validado.

Luego, llama al método *asyncData()* de la página para obtener y representar los datos en el lado del servidor antes de llamar al método *fetch()* para llenar el 'store' de Vuex en el lado del cliente.

En este punto, la página debe tener todos los datos y el contenido necesarios para poder mostrar una página web adecuada. Finalmente, el motor Nuxt.js procesa la página y la muestra al cliente, completando el proceso.

Este diagrama de flujo ilustra todos los pasos necesarios para representar una sola página:



4. Beneficios de Nuxt.js.

Cero configuraciones: Te permite empezar a desarrollar sin tocar configuraciones, también te facilita la configuración según sus necesidades de forma sencilla, siempre que desee modificar cualquiera de las configuraciones predeterminadas, simplemente se puede sobrescribirlo con el archivo *nuxt.config.js*

Enrutamiento automático: Vue.js requiere que configure la configuración del archivo router.js y coloque su enrutador allí, que se debe hacer manualmente. Nuxt.js lo libera de todo este trabajo al generar una configuración de enrutador automática, es decir, vue-router según los archivos .vue provistos dentro de la carpeta de páginas. Esto, a su vez, significa que ya no tiene que escribir configuraciones de enrutamiento. Para navegar entre páginas, puede usar el componente `<NuxtLink>`.

Modos de renderizado: Permite cambiar entre la generación de sitios estáticos o el renderizado de servidor bajo demanda.

Obtención de datos: Permite obtener contenido de cualquier fuente en los componentes Vue, listo para SSR (Server-Side Rendering).

Convenciones fuertes: Permite trabajar en equipos de manera eficiente gracias a una sólida estructura de directorios y convenciones.

Se tiene en cuenta el SEO (Search Engine Optimization): Permite definir etiquetas `<meta>` predeterminadas, como títulos específicos y etiquetas de descripción, para la aplicación dentro del archivo *nuxt.config.js* y proporciona tiempos más rápidos para indexación del contenido.

Importación automática de componentes: Nuxt ayuda a importar componentes de manera automática e inteligente.

Sistema de módulos: Implementa un sistema de módulos ofreciendo más de 160 módulos listos para importar y la posibilidad de crear tuyos propios.

5. Cómo crear una aplicación Nuxt.js.

Lo primero es ir a la [página oficial](#) al apartado de instalación. Te dan la opción de probar antes con dos opciones de playground online.

Antes de empezar deberás cumplir los requisitos. Tener node (recomendada la última versión), tener un editor de texto y tener una terminal.

Tenemos dos opciones para configurar el proyecto Nux. Manualmente o con el comando create-nuxt-app.

Para empezar de manera rápida se utilizará create-nux-app. El comando te pedirá algunas preguntas. Después de contestarlas instalará todas las dependencias y ya podrás ir a la carpeta del proyecto y lanzarlo. Con yarn dev se inicia en `http://localhost:3000`

```
yarn create nuxt-app <project-name>
```

```
cd <project-name>  
yarn dev
```

A partir de aquí ya puedes empezar con tu proyecto. Puedes crear enrutadores o middleware para la comunicación en tu proyecto. También puedes hacer uso de la librería de componentes Nuxt.js para crear componentes reutilizables y personalizables.

Si queremos depurar el código podemos usar el depurador de Chrome. Otra opción es usar el depurador en Visual Studio Code.

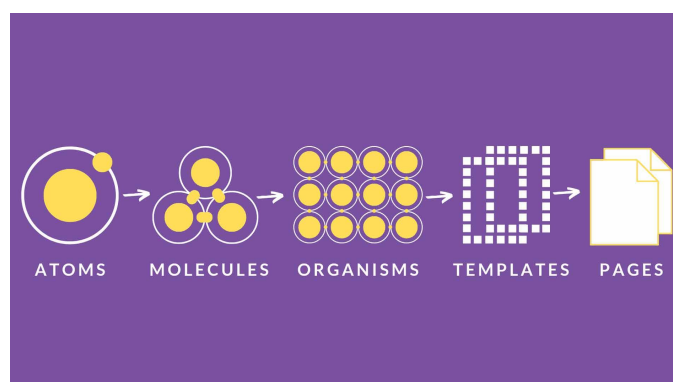
6. Reutilización y Atomic Web Design.

Nuxt nos permite realizar una gran variedad de componentes que podemos utilizar en nuestra aplicación. Es muy importante que estos componentes sean lo más óptimos posible, ya que para cada compilación el framework debe cargarlos. Para conseguir una gran optimización en nuestro desarrollo es importante que la gran mayoría de los componentes sean **reutilizables** y acoplables en nuestra aplicación, es decir, si desarrollamos un componente para un botón, o un input, este debe ser capaz de utilizarse mediante “props” en todos y cada uno de los ámbitos de la aplicación.

Por ejemplo, si continuamos con el ejemplo del botón, este debe tener propiedades cambiantes para modificar su texto, background, border, padding y demás propiedades.

Una de las características más importantes de esta metodología de estructura y diseño sería el mantenimiento de código, ya que, cuanto más reutilicemos nuestros componentes más sencillo será. Por ejemplo, imaginemos que estamos trabajando en una empresa en la que el equipo de diseño y UX/UI desde un principio planificó una paleta de colores y tonalidades verdes. Tras varios años de uso, deciden dar un cambio radical a la marca y pasar a tonalidades azuladas. Si nosotros no componetizamos y reutilizamos el elemento “button” en la web, habría que buscar uno por uno todos los de la aplicación y cambiar su color. Teniendo un único componente importado por el código, sólo tendríamos que cambiar el color de ese componente en específico.

Atomic Web Design nos ayuda a mejorar nuestra reutilización de forma sencilla y ágil. Presenta 5 niveles:



Nivel 1: Átomos

Aplicado al desarrollo, hablamos de átomos cuando hablamos de elementos que por sí solos tienen alguna funcionalidad: botones, *cards*, avatares, inputs de formulario...

Nivel 2: Moléculas

Una molécula es una unión de dos o más átomos. Por ejemplo, un campo de formulario con un input y un label indicativo.

Nivel 3: Organismos

Cuando llegamos al tercer nivel estamos hablando de elementos que son más complejos, ya que son sumas de los niveles anteriores.

Se tratan de elementos formados por moléculas (que a su vez son átomos) y que, por encima de todo, pueden repetirse. Podríamos considerar un organismo una unión de moléculas, átomos y otros organismos.

Nivel 4: Plantillas

Las plantillas vienen a ser los *wireframes*: una unión de distintos organismos que forman una página o una aplicación.

Nivel 5: Páginas

Este es el último nivel del *Atomic Design*. Son en realidad el diseño final que ya contiene las imágenes y otros detalles que hacen que ese archivo ya esté listo para el desarrollo.

Las ventajas de *Atomic Design* son:

- Facilita la creación de la guía de estilo.
- Permite hacer más rápido los prototipos, ya que los elementos ya existen.
- Hace más rápido el proceso de actualizar el diseño del producto y/o añadir nuevas funcionalidades, ya que los cambios no harán que se deba programar todo desde cero.
- Menos componentes hará el diseño y el código más consistentes y eficientes.
- Permite reutilizar átomos para crear cualquier diseño que se requiera.

Se adjunta con el trabajo un pequeño proyecto Nuxt, con una estructura Atomic Design y comentarios con explicación.

7. Bibliografía.

Nuxt.js - *Wikipedia* [en línea]. en.wikipedia.org.

Disponible en: <https://en.wikipedia.org/wiki/Nuxt.js>

Nuxt [en línea]. nuxtjs.org.

Disponible en: <https://nuxtjs.org/>

Nuxt - Installation [en línea]. nuxtjs.org. 10 de enero 2023

Disponible en: <https://nuxtjs.org/docs/get-started/installation>

Nuxt Lifecycle [en línea]. nuxtjs.org. 10 de enero 2023

Disponible en: <https://nuxtjs.org/docs/concepts/nuxt-lifecycle/>

¿Dónde encaja Nuxt.js en el desarrollo web? [en línea]. desarrollolibre.net, 25 de Enero 2022. Disponible en:

<https://www.desarrollolibre.net/blog/nuxt/donde-encaja-nuxtjs-en-el-desarrollo-web#:~:text=Su%20objetivo%20es%20crear%20aplicaciones,lugar%20al%20tratar%20con%20Nuxt>

What Is Nuxt.js? Learn More About the Intuitive Vue Framework [en línea].

kinsta.com. 23 de Noviembre 2022.

Disponible en: <https://kinsta.com/knowledgebase/nuxt-js/>