

TEXT CLASS REVIEW

TEMAS A TRATAR EN LA CUE

- ¿Qué es NuxtJs?
- ¿Qué funcionalidades tiene?

INTRODUCCIÓN

NuxtJS es un framework, que se utiliza para el desarrollo de aplicaciones web. Se puede emplear para crear aplicaciones estáticas (**static page**), de una sola página (**SPA**) o de servidor (**SSR**).

¿QUÉ ES NUXT?

Nuxt contiene toda la potencia que tienen **VueJS**, **Webpack** y **SSR**, y permite crear aplicaciones universales de una forma fácil, rápida y flexible, con menos configuración.

Si recordamos, las aplicaciones universales son aquellas que contienen módulos de código, capaces de ser ejecutados tanto en un navegador, como en un servidor. Por lo tanto, con **Nuxt** se podrá tener aplicaciones que cumplan con los requisitos de **SEO**, y de servir contenido al usuario lo antes posible, sin necesidad de renunciar a todo el dinamismo en cliente de las **SPAs** convencionales.

La ventaja de este framework, es que ya hace todo el trabajo engorroso de configuración y construcción por nosotros.

Su idea es permitirnos generar el **template** de un proyecto base, como hacíamos con **vue-cli**, pero con una estructura específica para trabajar con todos los elementos de **vue**, como: el **router**, los **stores**, y los componentes, de una manera más uniforme, más intuitiva, y con un sistema basado en convenciones y no tanto en configuraciones.

Por ahora, el proyecto se encuentra en versión Alpha, pero las expectativas puestas en él, por la comunidad **Vue**, son altas.

¿QUÉ FUNCIONALIDADES TIENE?

Puede que hasta ahora **Nuxt** nos parezca confuso, pero las funcionalidades que promete son de mucha utilidad. Entre ellas encontramos lo siguiente:

- Escritura de ficheros **Vue**: nada innovador. Una aplicación desarrollada en **Nuxt**, es lo mismo que una hecha en **Vue**. Por lo tanto, el mecanismo de generar nuestros componentes en ficheros de tipo **Vue** sigue igual. Puede que deseemos migrar un proyecto anterior de **Vue**, a la estructura de **Nuxt**, y ésta será asequible. Al final, **Nuxt**, se emplea para interpretar componentes **Vue** de una forma más limpia.
- Separación de código en paquetes de forma automática: **Webpack** y **Vue** nos ayudan a generar paquetes más pequeños, y que se cargan bajo demanda, para disminuir el principal y hacer nuestra aplicación más rápida. En **Nuxt**, no tendremos que hacer nada para activar este sistema, pues las vistas ya contienen esta separación en módulos dinámicos, y bajo demanda.
- Renderizado en la parte de servidor: sin configuraciones. Todas las vistas de la aplicación son renderizadas en servidor, y todos los cambios dinámicos son interceptados por **Vue** en cliente. **Nuxt** nos hace el **SSR** transparente.
- Sistema de rutas y sincronismo de datos avanzado: ya no es necesario configurar **vue-router**, pues **Nuxt** sabe exactamente qué rutas generar, dependiendo de cómo estructures tus vistas dentro del proyecto. Además, existe funcionalidad extra para la sincronización de datos y componentes.
- Servir ficheros estáticos: el propio proyecto será capaz de servir estáticos, ya sea de vistas, imágenes o fuentes; todo con un servidor integrado por defecto.
- Transpilación de **ES6/ES7**, preprocesamiento de **SASS/LESS/Stylus** y empaquetado/minificado de **JS** y **CSS**: todo el **Webpack** será configurado para que no sea necesario ocuparse de ello, solo de escribir código.
- Carga directa en desarrollo: **Nuxt** configura el proyecto, para que, en desarrollo, se acepte la carga inmediata de cambios.
- Generación de vistas a formato estático: es la funcionalidad más importante de **Nuxt**. Permite generar todo un proyecto **Vue** de manera estática, y renderizar todo en ficheros **HTML**, para que cualquier **CDN** pueda hospedarlo, y sea servido de una forma rápida y optimizada.
- Viene con una configuración lista para producción.