

TEXT CLASS REVIEW

TEMAS A TRATAR EN LA CUE

- Framework Quasar, y casos de uso.
- Características, ventajas y desventajas.
- Formas de instalar Quasar
- Setup de herramientas con Quasar CLI.

FRAMEWORK QUASAR, Y CASOS DE USO



QUASAR

BEYOND THE
FRAMEWORK

Con los avances de la tecnología, se empezaron a crear las aplicaciones para que se pudieran ver correctamente, en todos los distintos modelos de pantallas. Sin embargo, eso no era suficiente, ya que uno de los problemas que se presentaba, era la necesidad de desarrollar aplicaciones en distintos lenguajes y **frameworks**, como **Java** y **Swift**, para desplegarlos en las distintas plataformas (**Android**, **iOS**, entre otras), por lo que la tarea era bastante cara y tediosa. A raíz de ello, se incluyeron opciones de desarrollo multiplataforma, práctica que permite, bajo un mismo código, generar soluciones múltiples a la vez. Para el desarrollo en **Vue**, existe **Quasar**.

Quasar es un marco de trabajo basado en **Vue.js**, de código abierto con licencia del **MIT**, que permite a los desarrolladores web, crear rápidamente sitios o aplicaciones (**SPA**, **SSR**, **WPA**, extensiones de navegador, aplicaciones móviles y aplicaciones de escritorio). Se centra en facilitar todas las herramientas para la creación de aplicaciones muy visuales, y reducir el coste del desarrollo, permitiendo, con un mismo código, generar productos para diferentes plataformas.

CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS

Dispone de una colección completa de componentes basados en **Material Design**, y permite crear proyectos escribiendo un solo código, **SPA**, **SSR**, **PWA**, aplicaciones móviles híbridas, extensiones de navegador y, hasta aplicaciones de escritorio, a través de Electron, lo que ayuda a desarrollarlas en un tiempo récord. Posee una comunidad muy grande de usuarios. Cuenta con el mejor soporte de su clase para cada modo de compilación, y una buena experiencia para el desarrollador, a través de la estrecha integración con su propia **CLI**. Es fácilmente personalizable (**CSS**), y ampliable (**JS**). Tiene un ciclo de lanzamiento regular, que incluye nuevas funciones, creadas a partir de las sugerencias de su comunidad. Maneja toda la experiencia de desarrollo (incluida la creación de los iconos, y las pantallas de presentación de su aplicación). Es el marco más centrado en el rendimiento, y por ello, permite el uso de pruebas automatizadas, además, con su **SSR**, permite que mejore el **SEO** de Google.

Como desventaja, presentamos el caso de **iOS**, donde ellos tienen sus propios estilos de diseño, que no corresponden a los de **Material Design**, herramienta usada por **Quasar**.

FORMAS DE INSTALAR QUASAR

Podemos hacerlo de 3 formas diferentes:

- UMD / Standalone: incrustándolo en un proyecto existente a través de **CDN**, integración progresiva.
- Quasar CLI: creando un proyecto desde cero con la interfaz de línea de comandos de **Quasar**, es la única forma 100% segura de obtener el desarrollo multiplataforma deseado.
- Vue CLI: con la línea de comandos ofrecida de **Vue**, podemos integrar **Quasar** a una aplicación, aunque no se garantiza el soporte para aplicaciones multiplataformas, y adelanta probables inconvenientes con los **plugins**.

SETUP DE HERRAMIENTAS CON QUASAR CLI

Ahora que vimos las formas de integrar Quasar a nuestro proyecto, usaremos la más eficiente, que es a través de Quasar CLI. Para instalarla, necesitamos lo siguiente:

- Hardware: Preprocesador Intel I3 o similar mínimo, RAM 8GB mínimo.
- Software: Node JS 12.22.1; NPM 6.14.12