

Pautas para una buena componentización

Repositorio independiente por componente

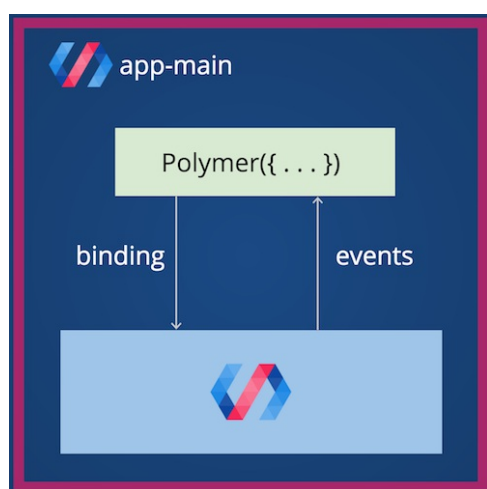
Cada componente debería tener un repositorio independiente para ser agregado como una dependencia en los archivos de configuración del proyecto principal, como 'package.json' o 'bower.json'.

Abstraete de la aplicación completa

Con el fin de construir un componente independiente y reutilizable, el desarrollador debe abstraerse de la aplicación completa y buscar la funcionalidad concreta que el componente tiene que completar y evitar el uso de múltiples dependencias.

Entiende el funcionamiento del flujo de datos

Tómate tu tiempo para entender cómo funciona el flujo de datos en Angular. Cada componente recibe propiedades por enlace de datos, usando atributos en la instanciación del componente padre. La comunicación entre los hijos y sus padres, debe utilizar los eventos del sistema nativo para una abstracción completa, como Angular EventEmitter.



Esto es muy importante, porque el uso de eventos para la comunicación evita las referencias directas entre los componentes.

Bloques funcionales completos e independientes

Busca bloques funcionales completos e independientes. Un botón único no es un buen componente, un botón con un campo de texto para validar / enviar / crear un campo es un buen componente. Un componente único tiene que dar una funcionalidad concreta al desarrollador, siguiendo el Principio de Responsabilidad Única. La granularidad del componente debe ser definida por el equipo de trabajo, para ser lo más útil posible.

Deben ser sin estado

Los componentes Web deben ser elementos sin estado, delegando esta administración de estado a un nivel superior en la jerarquía de componentes.

Mientras mas abstracción mejor

Es importante agregar tantas capacidades de abstracción como podamos, para facilitar la personalización de los componentes y su evolución. Por ejemplo, los desarrolladores deben crear una variable personalizada CSS3 para cada propiedad sensible al cambio, como colores de borde, color de texto o el relleno estándar del componente.

Componentiza con cuidado

No crees un componente para cada parte de tu aplicación. Simplemente crea un componente independiente y reutilizable cuando el elemento pueda ser reutilizado en otros contextos o aplicaciones. El exceso de componentes es el peor problema que se puede tener trabajando con una arquitectura orientada a componentes.

Características indispensables de un componente Web

- **Reutilizable:** el componente está listo para ser utilizado si el desarrollador lo importa desde `node_modules` / `bower_components`.
- **Independiente:** debe poder ejecutarse sin un servicio externo o dependencias.
- **Probado:** el componente tiene que ser probado y tener al menos el 60% de cobertura en su prueba de unidad.
- **Documentado:** el componente tiene que tener una documentación completa, generada con las convenciones actuales, para ser fácil de usar por los desarrolladores, presentando entradas / propiedades, métodos disponibles y reglas de estilo de personalización.