

¿Qué es Angular y cuáles son sus componentes clave?

Es una librería para interfaces de usuario basada en componentes, la lógica se escribe en typescript, y utiliza el “@component” decorator para convertir clases en componentes Angular. Las propiedades de estas clases se convierten en estados reactivos y se incorporan a la parte HTML de cada componente para refrescar la interfaz de usuario cada vez que se produce un cambio.

¿Cuál es la diferencia entre ngOnChanges y ngOnInit en un componente Angular?

Como su nombre lo indica, ngOnChanges vendría a ser un hook que actualiza sus valores al detectar cambios en un componente específico de la UI, mientras que ngOnInit sirve para realizar la configuración inicial del componente durante el proceso de renderizado.

Explique el concepto de "binding" en Angular y mencione los tipos de binding disponibles.

Es el proceso que permite intercambiar valores (o estados) de variables entre el layout HTML y el archivo typescript interactivo de configuración de un determinado componente.

- Está el binding bidireccional que permite actualizar los valores en HTML y también llamar los métodos que actualizan las variables en typescript
- Binding de propiedad que hace que una propiedad del HTML dependa de un valor específico en el typescript (pero no viceversa)
- Binding de evento para llamar los métodos del componente tras eventos en el HTML
- Y por último el de interpolación que es el que permite la reactividad.

¿Qué son los servicios en Angular y cómo se comunican con los componentes?

Por el contrario de los componentes, los servicios no renderizan directamente nada en la UI. Son clases que se encargan de permitir compartir estados entre componentes, implementar funciones que dependan de más de un componente y realizar operaciones que no se relacionan directamente con la UI (como fetch de datos por ejemplo). La principal manera de comunicarse con los componentes es mediante la inyección de dependencias.

Hable sobre las ventajas de Lazy Loading en una aplicación Angular y cuándo se debería utilizar.

La principal ventaja de usar Lazy Loading es mejorar el desempeño de los sitios web, principalmente en la carga inicial. Se debe utilizar en conjunto con el enrutamiento de Angular para cargar los componentes cuando el usuario navega a una página determinada, y no en la carga inicial de la aplicación.