[](https://angular.io/)

# Próximos pasos:

# herramientas y técnicas

Angular.io

Después de comprender los componentes básicos de Angular, puede comenzar a aprender más sobre las características y herramientas disponibles para ayudarlo a desarrollar y entregar aplicaciones Angular. Aquí hay algunas características clave.

## 

## Responsive programming

* [Ganchos de ciclo de vida](https://angular.io/guide/lifecycle-hooks) : aproveche los momentos clave en la vida útil de un componente, desde su creación hasta su destrucción, implementando las interfaces de gancho de ciclo de vida.
* [Observables y procesamiento de eventos](https://angular.io/guide/observables) : cómo utilizar observables con componentes y servicios para publicar y suscribirse a mensajes de cualquier tipo, como eventos de interacción del usuario y resultados de operaciones asincrónicas.

## 

## interacción cliente-servidor

* [HTTP](https://angular.io/guide/http) : comuníquese con un servidor para obtener datos, guardar datos e invocar acciones del lado del servidor con un cliente HTTP.
* [Representación del lado del servidor](https://angular.io/guide/universal) : Angular Universal genera páginas de aplicaciones estáticas en el servidor a través de la representación del lado del servidor (SSR). Esto le permite ejecutar su aplicación Angular en el servidor para mejorar el rendimiento y mostrar la primera página rápidamente en dispositivos móviles y de baja potencia, y también facilitar los rastreadores web.
* [Trabajadores de servicios](https://angular.io/guide/service-worker-intro) : utilice un trabajador de servicios para reducir la dependencia de la red y mejorar significativamente la experiencia del usuario.

## 

## bibliotecas específicas de dominio

* Animations : utilice la biblioteca de animación de Angular para animar el comportamiento de los componentes sin un conocimiento profundo de las técnicas de animación o CSS.
* Forms : admite escenarios de entrada de datos complejos con validación basada en HTML y verificación sucia.

## Soporte para el  ciclo de desarrollo

* [Compilación](https://angular.io/guide/aot-compiler) : Angular proporciona una [compilación](https://angular.io/guide/aot-compiler) justo a tiempo (JIT) para el entorno de desarrollo y una compilación anticipada (AOT) para el entorno de producción.
* [Plataforma de prueba](https://angular.io/guide/testing) : ejecute pruebas unitarias en las partes de su aplicación mientras interactúan con el marco angular.
* [Internacionalización](https://angular.io/guide/i18n) : haga que su aplicación esté disponible en varios idiomas con las herramientas de internacionalización de Angular (i18n).
* [Pautas de seguridad](https://angular.io/guide/security) : conozca las protecciones integradas de Angular contra vulnerabilidades y ataques comunes de aplicaciones web, como los ataques de secuencias de comandos entre sitios.

## 

## configuración de instalación, compilación e implementación

* [Referencia de comandos de la CLI](https://angular.io/cli) : la CLI angular es una herramienta de línea de comandos que se utiliza para crear proyectos, generar código de aplicación y biblioteca y realizar una variedad de tareas de desarrollo continuas, como pruebas, agrupación e implementación.
* [Espacio de trabajo y estructura de archivos](https://angular.io/guide/file-structure) : Comprenda la estructura del espacio de trabajo angular y las carpetas de proyectos.
* [Paquetes npm](https://angular.io/guide/npm-packages) : Angular Framework, Angular CLI y los componentes utilizados por las aplicaciones [Angulares](https://docs.npmjs.com/) se empaquetan como paquetes [npm](https://docs.npmjs.com/) y se distribuyen a través del registro npm. La CLI angular crea un package.jsonarchivo predeterminado , que especifica un conjunto inicial de paquetes que funcionan bien juntos y soportan conjuntamente muchos escenarios de aplicación comunes.
* [Configuración de TypeScript](https://angular.io/guide/typescript-configuration) : TypeScript es el idioma principal para el desarrollo de aplicaciones angulares.
* [Soporte de navegador](https://angular.io/guide/browser-support) : haga que sus aplicaciones sean compatibles en una amplia gama de navegadores.
* [Creación y servicio](https://angular.io/guide/build) : aprenda a definir diferentes configuraciones de compilación y servidor proxy para su proyecto, como desarrollo, preparación y producción.
* [Implementación](https://angular.io/guide/deployment) : aprenda técnicas para implementar su aplicación Angular en un servidor remoto.