

# ANGULAR UNIVERSAL

Angular



# ANGULAR

Introducción y ventajas



# ANGULAR

- ¿Qué es Angular Universal?
  - Es la tecnología de Angular para implementar SSR (Server Side Rendering) y SSG (Static Site Generation o *prerendering*).
    - SSR → Proporciona una versión renderizada en el servidor de la aplicación (en cada petición). En segundo plano, descarga la versión “normal” de la aplicación. Cuando la descarga de esta última ha terminado, se sustituye la versión renderizada por la otra.
    - SSG → Renderiza algunas páginas durante el *build* generando archivos HTML estáticos listos para desplegar.
  - Antes de la versión 17 era un paquete separado (el proyecto Angular Universal, ya obsoleto).
  - A partir de la versión 17, está integrado en CLI y en el paquete **@angular/ssr**.



# ANGULAR

- Ventajas de SSR y SSG:
  - Reduce los tiempos de carga.
  - Reduce la carga de trabajo en el cliente.
  - Mejora el posicionamiento ya que permite la aplicación de las técnicas de SEO.
  - En ocasiones, puede permitir que la aplicación sea accesible a través de un navegador sin soporte para JavaScript.
  - (SSG) → Reduce la carga de trabajo en el servidor.



# ANGULAR

- Creación del proyecto:
  - Opción 1: Seleccionando ‘y’ en la pregunta “*Do you want to enable Server-Side Rendering (SSR) and Static Site Generation (SSG/Prerendering)? (y/N)*”
  - Opción 2: Agregando Angular SSR sobre un proyecto existente: `ng add @angular/ssr`



# ANGULAR

- **El proyecto SSR contiene algunos archivos nuevos:**

- `server.ts` → Actúa como el servidor web real (generalmente basado en Express) que maneja las solicitudes HTTP. Su función es recibir las peticiones del usuario y utilizar el bundle de servidor de Angular (el código de `main.server.ts`) para responder con el HTML renderizado.
- `assets/main.server.ts` → Es el punto de entrada (entry point) de la aplicación Angular cuando se ejecuta en el servidor. Su función principal es arrancar la aplicación Angular en un entorno sin navegador.



# ANGULAR

## ■ Ejecución del proyecto:

The screenshot shows the Chrome DevTools Console tab. The tabs at the top are Elements, Console (which is selected), Sources, Network, Performance, Memory, Application, and Privacy and security. Below the tabs, there are buttons for play/pause, stop, and refresh, followed by dropdown menus for 'top' and 'Filter'. The main content area displays the following text:  
Angular is running in development mode.  
Angular hydrated 1 component(s) and 71 node(s), 0 component(s) were skipped. Learn more at <https://angular.dev/guide/hydration>.  
> `ctrl i` to turn on code suggestions. [Don't show again](#) NEW

Con SSR

The screenshot shows the Chrome DevTools Console tab. The tabs at the top are Elements, Console (which is selected), Sources, Network, Performance, and Memory. Below the tabs, there are buttons for play/pause, stop, and refresh, followed by dropdown menus for 'top' and 'Filter'. The main content area displays the following text:  
Angular is running in development mode.  
> `ctrl i` to turn on code suggestions. [Don't show again](#) NEW

Con CSR



# ANGULAR

- Tamaño de la descarga:

- Se puede obtener con el comando **curl**:

- curl -get http://localhost:4200/ -o ssr.txt

- Ejemplo:

- 28/11/2025 09:27 457 csr.txt
    - 28/11/2025 09:28 22.504 ssr.txt



# ANGULAR

- **Construcción del proyecto:**
  - La compilación del proyecto con `ng build --configuration production` genera dos carpetas dentro de la carpeta `dist/nombre-del-proyecto`:
    - `browser` → código de distribución de la aplicación CSR (proyecto normal).
    - `server` → código de distribución de la aplicación SSR.
- **Para ejecutar la aplicación SSR se requiere un servidor Node:**
  - `node ssr-demo\server\server.mjs`



# ANGULAR

- Indicando el tipo de renderizado en el routing:
  - En el fichero app.routes.server.ts de la aplicación se puede indicar qué técnica de renderizado aplicar a cada “ruta”:

## Rendering modes

The server routing configuration lets you specify how each route in your application should render by setting a `RenderMode`:

Rendering mode	Description
Server (SSR)	Renders the application on the server for each request, sending a fully populated HTML page to the browser.
Client (CSR)	Renders the application in the browser. This is the default Angular behavior.
Prerender (SSG)	Prerenders the application at build time, generating static HTML files for each route.



# ANGULAR

## ■ Enlaces de interés:

- <https://angular.dev/guide/ssr>
- <https://blog.angular-university.io/angular-universal/>
- ¿Por qué creo todos mis proyectos con SSR en Angular? (YouTube)
  - [https://youtu.be/kvDhx1B\\_rP4?si=IrXBGHMC1cUN4Xjt](https://youtu.be/kvDhx1B_rP4?si=IrXBGHMC1cUN4Xjt)