



## Guía de migración de aplicaciones angular de 8.x.x a 15.x.x

---

8 de Septiembre de 2023

# Índice

Proceso de Migración.....	3
Antes de migrar: .....	3
Angular 8 a 9 .....	3
Angular 9 a 10 .....	4
Angular 10 a 11.....	4
Angular 11 a 12.....	4
Angular 12 a 13.....	5
Angular 13 a 14.....	5
Angular 14 a 15.....	5

## Proceso de Migración

Para el proceso de migración, seguiremos la guía disponible en la siguiente URL: <https://update.angular.io/?l=3&v=8.2-15.0>. Asegúrate de marcar la opción de Angular Material si se está utilizando en el proyecto.

### Antes de migrar:

Asegúrese de crear una rama limpia del proyecto que desea migrar (sin commits por subir o cambios por guardar).

Los comandos genéricos que vamos a utilizar en todos los procesos de migración son:

- `cmd /C "set "NG_DISABLE_VERSION_CHECK=1" && npx @angular/cli@X update @angular/cli@X @angular/core@X"` substituyendo la X siempre por el número de versión al que nos migramos
- Si el proyecto utiliza angular material también usaremos `cmd /C "set "NG_DISABLE_VERSION_CHECK=1" && npx @angular/cli@X update @angular/material@X"` substituyendo también la X siempre por el número de versión al que nos migramos

### Angular 8 a 9

1. Modifica el `package.json` eliminando las dependencias de `Ontimize` y realiza una instalación (`npm install`) (**node 12.22.12**).
2. Ejecuta los comandos de migración de acuerdo con la guía. Asegúrate de que el repositorio esté limpio, con cambios commiteados, stageados o subidos.
  - `cmd /C "set "NG_DISABLE_VERSION_CHECK=1" && npx @angular/cli@8 update @angular/cli@8 @angular/core@8"` (**node 16.14.0**)
  - `cmd /C "set "NG_DISABLE_VERSION_CHECK=1" && npx @angular/cli@9 update @angular/cli@9 @angular/core@9"` (**node 16.14.0**)
3. Después de ejecutar el comando, comprueba los siguientes cambios realizados:
  - Eliminación de `hammerjs`, si existe, tanto como dependencia como referencias.
  - En `tsconfig.json`, ajusta el `target` a "ES2015".
  - Los `ViewChild` no pueden tener la propiedad `static`.
4. La dependencia de `@angular/flex-layout` no se actualizará automáticamente. Debes cambiar la versión manualmente en este caso: `@angular/flex-layout": "~9.0.0-beta.31`.
5. Si el proyecto utiliza angular material se deberá ejecutar el siguiente comando `cmd /C "set "NG_DISABLE_VERSION_CHECK=1" && npx @angular/cli@9 update @angular/material@9"`

## Angular 9 a 10

1. Sigue los pasos anteriores para ejecutar la migración a Angular 10.
2. Modifica `tsconfig.json`: el `module` debe ser "es2020" y el `target` debe ser "es2018".
3. Actualiza la dependencia de `@angular/flex-layout` a la versión: `^10.0.0-beta.32`.

`npx @angular/cli@10 update @angular/core@10 @angular/cli@10 (node 16.14.0)`

## Angular 10 a 11

1. Sigue los pasos anteriores para ejecutar la migración a Angular 11.
2. Realiza los siguientes cambios:
  - Reemplaza `async` con `waitForAsync`.
  - Si en `app-routing` hay un `import` como este: `imports: [RouterModule.forRoot(routes, opt)]`, el objeto `opt` debe estar tipado y ser de tipo `ExtraOptions`.
3. Actualiza la dependencia de `@angular/flex-layout` a la versión: `^11.0.0-beta.33`.
4. `ng update @angular/core@11 @angular/cli@11`
5. Recuerda actualizar angular material si existiese en el proyecto

## Angular 11 a 12

1. Sigue los pasos anteriores para ejecutar la migración a Angular 12.
2. Realiza los siguientes cambios:
  - Modifica `angular.json` añadiendo `scripts` y elimina `aot` que debe estar en `true`.
  - Actualiza la importación de `zone` en archivos como `polyfills`, `environment.ts`, solo importa de `zone.js`, no de `dist/zone`.
  - Elimina la opción `emitDecoratorMetadata` en `tsconfig.json`.
  - Realizar el proceso de migración de `node-sass` a `sass`(<https://sass-lang.com/documentation/cli/migrator/>)
  - Se reemplaza dependencia de `node-sass` por `sass`
  - `script "npx node-sass ./src/assets/themes/ -o ./src/assets/themes/ --recursive" por npx sass --no-source-map --load-path=node_modules/ ./src/assets/themes:./src/assets/themes`
  - En los temas, agrega el uso de Angular Material: `@use '~@angular/material' as mat`.
3. Actualiza la dependencia de `@angular/flex-layout` a la versión: `^12.0.0-beta.34`.
4. Recuerda actualizar angular material si existiese en el proyecto

## Angular 12 a 13

1. Sigue los pasos anteriores para ejecutar la migración a Angular 13.
2. Realiza los siguientes cambios:
  - Añade la caché de Angular, que se puede deshabilitar, y agrégalo a `.gitignore`.
  - Elimina `lint` y `sourcemap` de `angular.json`.
  - En `test.ts`, `platformBrowserDynamicTesting` debe ser modificado con:

```
platformBrowserDynamicTesting(), {  
  teardown: { destroyAfterEach: false }  
}
```
  - En `tsconfig`, agrega la opción `"allowSyntheticDefaultImports": true`, ajusta el `target` a `"es2020"` y en `lib` también `"es2020"`.
  - En los temas, cambia los `use` de material a: `@use '@angular/material' as mat`.
3. Actualiza la dependencia de `@angular/flex-layout` a la versión: `^13.0.0-beta.38`.
4. Recuerda actualizar angular material si existiese en el proyecto

## Angular 13 a 14

1. Sigue los pasos anteriores para ejecutar la migración a Angular 14.
2. Realiza los siguientes cambios:
  - Modifica `angular.json` eliminando la opción `"defaultProject"`.
  - Añade las siguientes dependencias:

```
"@typescript-eslint/eslint-plugin"  
"@typescript-eslint/parser"
```
3. Actualiza la dependencia de `@angular/flex-layout` a la versión: `^14.0.0-beta.40`.
4. Recuerda actualizar angular material si existiese en el proyecto

## Angular 14 a 15

1. Sigue los pasos anteriores para ejecutar la migración a Angular 15.
2. Realiza los siguientes cambios:
  - Modifica `.browserslistrc` con:

```
last 2 Chrome versions  
last 1 Firefox version  
last 2 Edge major versions  
last 2 Safari major versions  
last 2 iOS major versions
```
  - Añade `"useDefineForClassFields": false` en `tsconfig.json`.
  - Añade las importaciones de `OnTimize`, excepto la dependencia de `theming`.
  - Modifica las importaciones de los temas, que ahora vienen directamente del core.
  - Puede que aparezcan advertencias de dependencias que deben añadirse a `"allowedCommonJsDependencies"` en `angular.json`.

**Importante:** En `app.module`, ya no se importa `OnTimizeModules`, ahora debes importar los módulos uno a uno, por ejemplo:

```
BrowserModule,  
BrowserAnimationsModule,  
OnTimizeWebModule.forRoot(CONFIG),
```

3. Actualiza la dependencia de `@angular/flex-layout` a la versión: `^15.0.0-beta.42`.
4. Recuerda actualizar angular material si existiese en el proyecto

Recuerda que este proceso puede ser complejo y puede haber otros ajustes necesarios en función de las particularidades de tu proyecto. Siempre es recomendable realizar pruebas exhaustivas después de cada paso de migración.