CREACIÓN DE LIBRERÍAS DE COMPONENTES

Angular

ANGULAR

Introducción a las librerías de componentes

ANGULAR

- Una librería en Angular es un conjunto de componentes, directivas, pipes y servicios reutilizables empaquetados para ser usados en uno o varios proyectos Angular.
- Ejemplo: Angular Material.
- Distribuible a través de npm (crear una cuenta gratuita)

ANGULAR

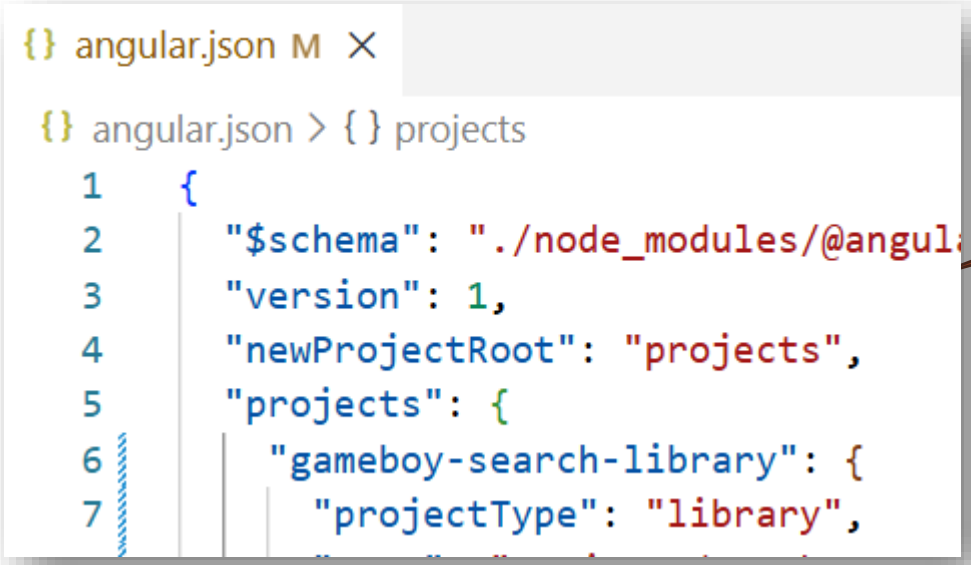
- Creación de un proyecto específico para una librería:
  - ng new mi-workspace --create-application=false
- Creación de una librería dentro de un proyecto:
  - ng generate **library** *mi-librería*
  - → Genera los ficheros en la carpeta **projects/*mi-libreria/***

Si se quiere crear el proyecto a posteriori:
ng generate application *nombre-app*

```
CREATE projects/mi-libreria/ng-package.json (166 bytes)
CREATE projects/mi-libreria/package.json (227 bytes)
CREATE projects/mi-libreria/README.md (1518 bytes)
CREATE projects/mi-libreria/tsconfig.lib.json (531 bytes)
CREATE projects/mi-libreria/tsconfig.lib.prod.json (412 bytes)
CREATE projects/mi-libreria/tsconfig.spec.json (430 bytes)
CREATE projects/mi-libreria/src/public-api.ts (85 bytes)
CREATE projects/mi-libreria/src/lib/mi-libreria.spec.ts (580 bytes)
CREATE projects/mi-libreria/src/lib/mi-libreria.ts (225 bytes)
UPDATE angular.json (3411 bytes)
UPDATE package.json (1167 bytes)
UPDATE tsconfig.json (1260 bytes)
√ Packages installed successfully.
```

ANGULAR

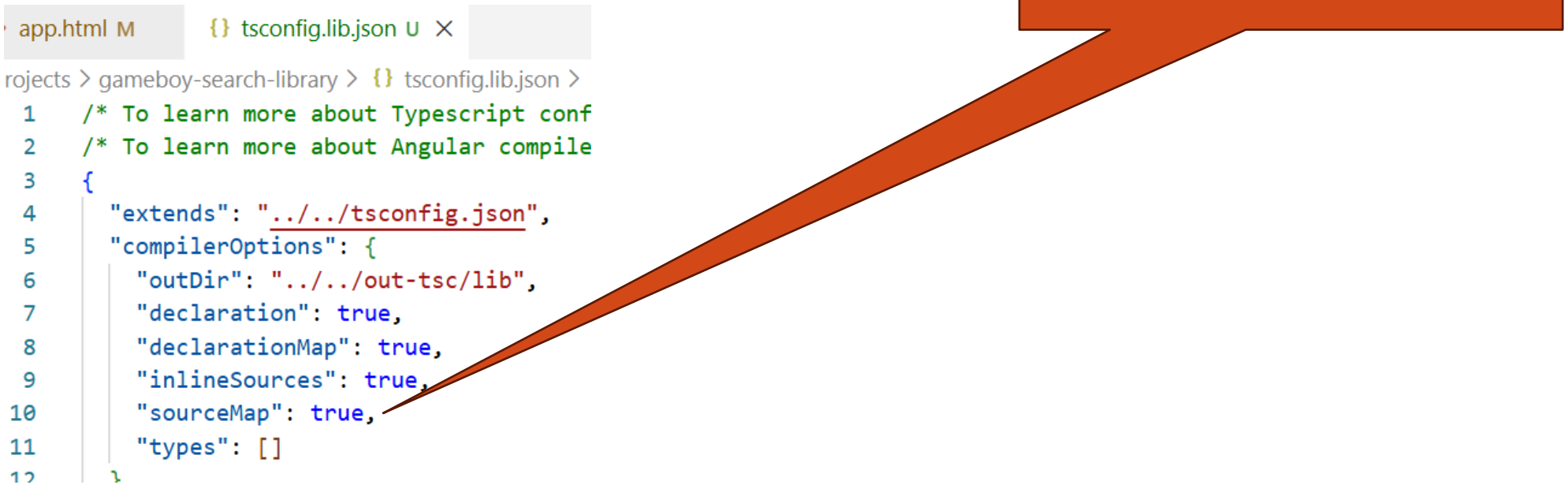
- Creación de una librería dentro de un proyecto:



Se añade automáticamente a angular.json

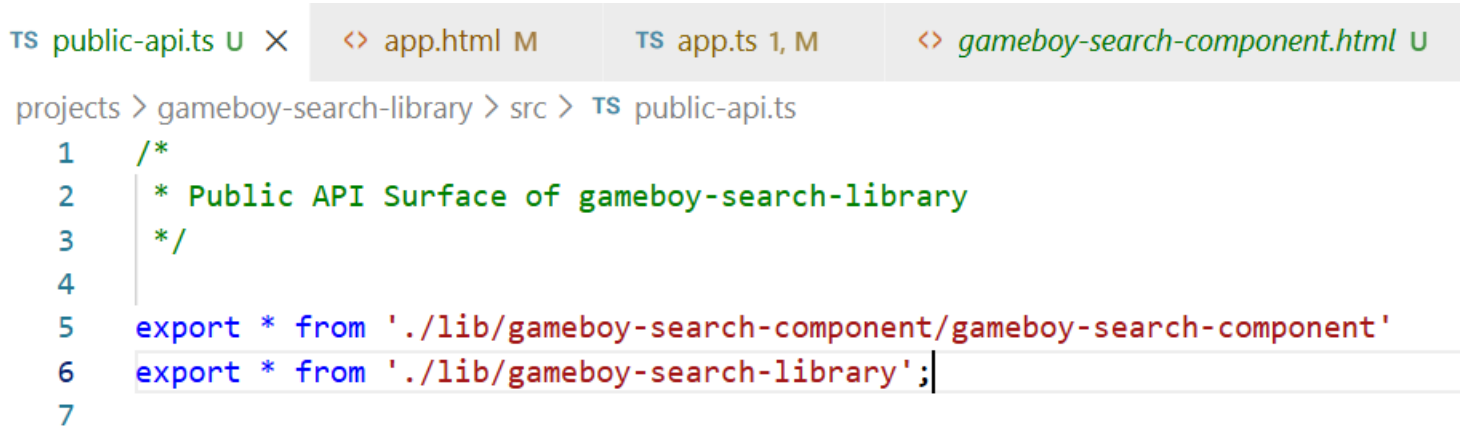
ANGULAR

- Si se utiliza en el propio proyecto, se debe agregar el atributo “sourceMap”:true al fichero tsconfig.lib.json



ANGULAR

- Crear componentes dentro de la librería:
  - ng generate component *nombre-componente* --project=*nombre-librería*
- Exportar componentes (fichero *public-api.ts*).
  - El fichero *public-api.ts* contiene las exportaciones de los componentes de la librería para que estos sean públicos.



- Importar componentes:
  - En los ficheros ts de los componentes donde se vayan a utilizar.

ANGULAR

Hay que hacer el build después de los cambios

Los cambios se actualizan de forma automática

- Compilación de la librería:
  - **ng build** *nombre-librería*
  - **ng build** *nombre-librería* **--watch**
  - El resultado se encuentra en dist/*nombre-librería*
- Utilización en el mismo proyecto:
  - **npm install** ./dist/*nombre-libreria*
- Distribución local:
  - Empaquetado:
    - Desde la carpeta dist/*nombre-librería*:
      - npm pack → Genera un fichero tgz
  - Instalación en el nuevo proyecto:
    - npm install *path/fichero.tgz*

ANGULAR

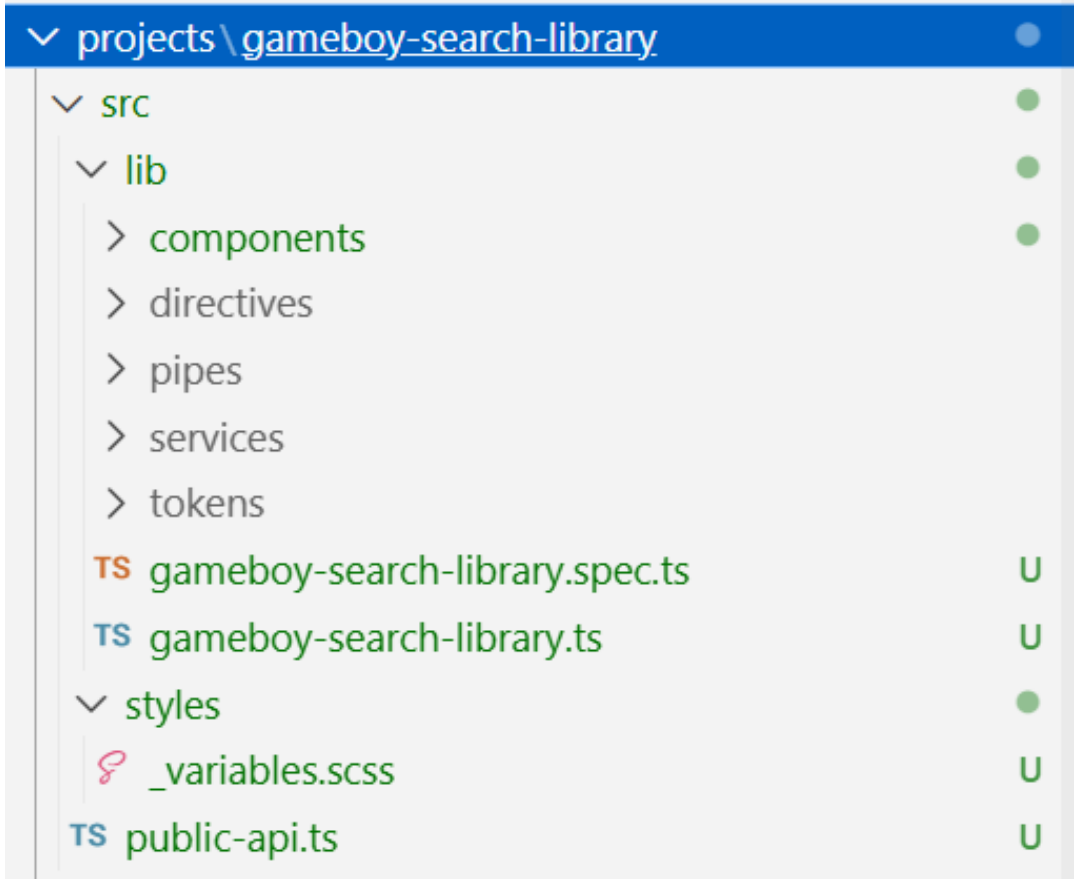
Buenas prácticas

ANGULAR

- Buenas prácticas:
  - Establecer una buena organización.
  - Uso de componentes *standalone*.
  - Centralización de la exportación a partir del fichero *public-api.ts*
  - Separar lógica y presentación.
  - Documentación.
  - Testing.
  - Optimizar dependencias externas.

ANGULAR

- Buenas prácticas:
- 1. Arquitectura base.
  - librería
    - src
      - lib
        - Componentes.
        - Directivas personalizadas.
        - Pipes personalizadas.
        - Servicios.
        - Tokens de inyección.



ANGULAR

- Organizar los componentes por grupos utilizando *entry point*.
  - Un ***entry point*** es un punto de entrada público a una librería.
  - Es el archivo que exporta los componentes, servicios y módulos que se ofrecen a los consumidores de la librería.
  - **public-api.ts** es el **entry point** principal.
  - Los ***entry points*** se pueden organizar en grupos, para facilitar su importación y mejorar el ***tree-shakin.***

Técnica de optimización para eliminar el "código muerto" en los *bundles* de distribución de la aplicación.

ANGULAR

- Cada componente tiene su propio publi-api.ts.
- La librería tiene un public-api.ts global (opcional, solo si se quiere permitir la importación de todos los elementos de la librería).

```
mi-libreria/
├── src/
│   ├── **componente-a/**
│   │   ├── componente-a.ts       // Lógica principal
│   │   ├── modelos.ts            // Tipos/Interfaces de A
│   │   └── **public-api.ts** // 👈 Punto de entrada Secundario para A
│   │
│   ├── **componente-b/**
│   │   ├── componente-b.ts       // Lógica principal
│   │   └── **public-api.ts** // 👈 Punto de entrada Secundario para B
│   │
│   ├── **componente-c/**
│   │   ├── componente-c.ts       // Lógica principal
│   │   ├── utilidades.ts         // Funciones de ayuda
│   │   └── **public-api.ts** // 👈 Punto de entrada Secundario para C
│   │
│   └── **public-api.ts** // 👈 Punto de entrada **Principal** (Exporta A, B, y C)
│
├── package.json
└── ...
```

ANGULAR

Storybook

ANGULAR

- Creación de *Storybook*.
  - ***Storybook*** es una herramienta para poder probar los componentes de manera independiente, obligando a mejorar el diseño y aumentando la productividad del desarrollo.
  - Instalación:
    - `npx storybook init`

ANGULAR

Optimización del *build* y *lazy loading*

ANGULAR

- Optimización en la construcción del build:
  - ng build --configuration=production
  - Habilita AOT (precompilación), Minificación y *Tree Shaking*
  - Se encarga de paquetizar el proyecto y verificar la configuración recogida en angular.json:

```
"configurations": {
            "production": {
              "budgets": [
                {
                  "type": "initial",
                  "maximumWarning": "500kB",
                  "maximumError": "1MB"
                },
```

ANGULAR

- Optimización programática mediante *lazy loading:*
- El Lazy Loading (carga diferida) en Angular es una técnica de optimización de rendimiento que permite cargar partes de la aplicación solo cuando son necesarias, en lugar de cargar todo el código JavaScript al inicio.
  - Uso en routing
  - Mediante @defer

ANGULAR

- Enlaces de interés:
  - Angular libraries:
    - https://angular.dev/tools/libraries/creating-libraries
  - Angular build:
    - https://angular.dev/tools/cli/build
  - Lazy Loading:
    - https://medium.com/@neelendra1destiny/evolution-of-lazy-loading-in-angular-19-with-standalone-components-fadd3b34603f
  - Storybook:
    - https://storybook.js.org/