



Modo IA

Todo

Imágenes

Vídeos

Noticias

Más ▾



en angular, debo crear los componentes dentro del directorio app o recomiendas otro directorio



Para 2025, la recomendación técnica estándar en **Angular (v17+)** es no mezclar todos los componentes directamente en la raíz del directorio `app/`. Aunque técnicamente funcionan allí, para que tu proyecto sea escalable y mantenable, lo ideal es organizar el código basándose en **características (features)** y reutilización.

Esta es la estructura de directorios recomendada para 2025:

1. Directorios sugeridos dentro de `app/`

- `features/`: Crea una subcarpeta por cada sección lógica de tu aplicación (ej: `auth`, `products`, `dashboard`). Cada carpeta debe contener sus propios componentes, servicios y rutas específicos.
- `shared/`: Para componentes puramente visuales y sin lógica de negocio que usarás en muchas partes (ej: un botón personalizado, un spinner o un modal genérico).
- `core/`: Reservado para servicios globales únicos (singletons) como la gestión de autenticación, interceptores o configuración global.
- `layout/`: Componentes que definen la estructura visual fija, como el `header`, `footer` o el `sidebar`.

2. Uso de Componentes Standalone (Recomendado)

Desde 2024 y consolidado en 2025, lo estándar es usar **componentes independientes (standalone)**. Estos eliminan la necesidad de archivos `*.module.ts` pesados, permitiendo que cada componente gestione sus propias

Pregunta lo que quieras





Modo IA

Todo

Imágenes

Vídeos

Noticias

Más ▾



```
└── layout/           # NAVBar, Footer
    └── features/      # Lógica de negocio
        ├── products/
        │   ├── components/
        │   ├── pages/     # Componentes con ruta propia
        │   ├── services/
        │   └── products.routes.ts
    └── users/
```

Usa el código con precaución.



¿Cómo crearlos con el CLI?

Para mantener este orden, usa el comando especificando la ruta. Por ejemplo:

- Para un componente de una característica: `ng g c features/nombre-feature/components/nombre-componente`.
- Para un componente compartido: `ng g c shared/components/mi-botón`.

Conclusión: No los dejes en `app/`. Organízalos por **features** para facilitar el *lazy loading* (carga perezosa) y mejorar el rendimiento de tu aplicación.

Las respuestas de la IA pueden contener errores. [Más información](#)





Modo IA

Todo

Imágenes

Vídeos

Noticias

Más ▾

