# Capítulo 04: Estructura del proyecto Next.js (Parte 1: Raíz)

## Carpetas en la raíz

### 1) .next/

Carpeta generada por Next.js cuando corrés `pnpm dev` o `pnpm build`. Guarda artefactos de compilación y caché.  
¿Se toca? No. Se borra sola cuando corresponde. Si algo anda raro, podés hacer `pnpm clean` (o borrar y volver a compilar).

### 2) node\_modules/

Todas las dependencias instaladas (Next, React, etc.).  
¿Se toca? No. Nunca editar a mano. Se recrea con `pnpm install`.

### 3) public/

Archivos estáticos que se sirven “tal cual” (imágenes, íconos, pdfs).  
Un archivo `public/next.svg` se accede en `/next.svg`.  
¿Se toca? Sí, cuando agregás assets.

### 4) app/

El corazón del enrutado (App Router).  
Regla clave: cada subcarpeta define una ruta y necesita un `page.js`.  
¿Se toca? Sí (lo veremos en la Parte 2).

## Archivos en la raíz

### 5) .gitignore

Lista lo que no se sube a Git (ej. `node\_modules/`, `.next/`).  
¿Se toca? Rara vez, si querés agregar exclusiones.

### 6) eslint.config.mjs

Configuración flat de ESLint (formato moderno en Next 15).  
Sirve para reglas de estilo/errores de JS/React.  
¿Se toca? A veces, si querés ajustar reglas del linter.

### 7) jsconfig.json

Configuración de paths/alias para imports.  
Import alias por defecto: `@/\*` apunta a la raíz del proyecto.  
Ejemplo: `import Button from "@/components/Button"`.  
¿Se toca? Casi nunca. Solo si querés cambiar/añadir alias.

### 8) next.config.mjs

Configuración avanzada de Next (imágenes remotas, i18n, etc.).  
¿Se toca? Solo cuando necesitás una opción específica. Al inicio, lo dejás como está.

### 9) package.json

Manifiesto del proyecto (scripts + dependencias).  
Scripts típicos:  
- `dev` → servidor de desarrollo  
- `build` → compila para producción  
- `start` → corre el build  
- `lint` → ejecuta ESLint  
¿Se toca? Sí, cuando sumás librerías o querés agregar scripts.

### 10) pnpm-lock.yaml

Candado de versiones para instalar exactamente lo mismo en todas las máquinas.  
¿Se toca? No a mano. Se actualiza solo al instalar/actualizar paquetes.

### 11) postcss.config.mjs

Integra Tailwind a través del plugin `@tailwindcss/postcss`.  
¿Se toca? Casi nunca. Suele quedar como viene.

### 12) README.md

Notas del proyecto (cómo correr, scripts, etc.).  
¿Se toca? Opcional, para documentar instrucciones propias.

## Nota sobre Tailwind

No ves `tailwind.config.js` porque estás con Tailwind 4.x: muchas cosas vienen autoconfiguradas.  
Si después querés personalizar (colores/temas/plugins), lo generás con:  
 pnpx tailwindcss init