**🧩 Nivel 1 – Fundamentos (Componentes, Data Binding, Directivas)**

👉 Objetivo: entender cómo funciona Angular por dentro.

1. **Lista de tareas (To-Do App)**
   * Crear, editar, eliminar y marcar tareas como completadas.
   * Usar @Input y @Output para comunicación entre componentes.
   * Aplicar directivas (\*ngFor, \*ngIf) y pipes (date, uppercase, etc.).
2. **Contador interactivo**
   * Incrementar/decrementar valores con botones.
   * Practicar el uso de ngModel y two-way binding.
3. **Mini portafolio personal**
   * Mostrar tu nombre, skills y proyectos.
   * Enrutamiento simple entre secciones (home, sobre mí, contacto).

**📡 Nivel 2 – Servicios, Routing y API REST**

👉 Objetivo: aprender arquitectura modular y consumo de datos.

1. **Libro de direcciones / Agenda de contactos**
   * CRUD completo conectado a una API REST.
   * Uso de HttpClient, Observables y servicios (@Injectable).
   * Rutas protegidas para crear/editar contactos.
2. **Buscador de películas o Pokémon**
   * Llamadas HTTP a una API pública.
   * Búsqueda en tiempo real con RxJS (debounceTime, switchMap).
   * Componente de detalle dinámico usando rutas con parámetros.

**🗂️ Nivel 3 – Formularios, Validaciones y Autenticación**

👉 Objetivo: manejar formularios complejos y seguridad.

1. **Formulario de registro y login con Firebase o backend propio**
   * Formularios reactivos con validaciones (FormBuilder, Validators).
   * Guardado de usuarios en Firebase o API Node.js.
   * Manejo de sesiones con localStorage o JWT.
2. **Dashboard con autenticación protegida**
   * Acceso a rutas sólo si el usuario está logueado (AuthGuard).
   * Cierre de sesión y control de expiración del token.

**🧠 Nivel 4 – Estado Global, Observables Avanzados y Arquitectura**

👉 Objetivo: pensar como un desarrollador profesional.

1. **Carrito de compras (E-Commerce básico)**
   * Añadir productos, actualizar cantidades y calcular total.
   * Estado global con BehaviorSubject o librerías como NgRx.
   * Paginación y filtros dinámicos.
2. **Panel de administración CRUD con roles**
   * Usuarios con permisos diferentes (admin, editor, lector).
   * Guardias basados en roles.
   * Lazy loading de módulos para mejorar el rendimiento.

**🌐 Nivel 5 – Proyecto Final (Full Stack)**

👉 Objetivo: unir todos los conocimientos en un proyecto real.

1. **Aplicación completa (ej. Blog o Gestión de tareas)**

* Backend en Node.js/NestJS o Firebase.
* CRUD completo con autenticación y roles.
* Formularios complejos, rutas anidadas, carga diferida, guards.
* Deployment en Vercel o Firebase Hosting.

💡 **Tips para aprender más rápido:**

* Usa **Angular CLI** en todos los proyectos para aprender las convenciones.
* Crea **servicios reutilizables** y separa el código en módulos.
* Trabaja con **interfaces y tipos en TypeScript** desde el inicio.
* Añade siempre un archivo README.md para documentar cada proyecto.