

Creamos la carpeta

En Visual studio Code abrimos terminal y ejecutamos el siguiente código `npm install -g @angular/cli`

Creamos el proyecto: con el siguiente comando `ng new zoo-app` estilo Tailwind CSS, No, None

Levantamos el proyecto:

```
cd zoo-app
```

```
ng serve
```

creamos modelo interfaces

```
ng generate interface models/animal
```

Crear directorio de modelos:

```
mkdir -p src/app/models
```

en model ponemos la interface

```
export interface Animal {  
  id_entidad: string;    // Usamos string para el ID que manejas  
  nombre: string;  
  especie: string;      // Cambiamos 'email' por 'especie'  
  fecha_nacimiento: Date;  
  email_cuidador: string; // Adaptando el campo email  
  clave: string;        // Aunque no la uses, suele venir en el examen  
  fecha_registro: Date;  
}
```

Creamos componentes

```
ng generate component componentes/animal
```

en componentes en animal.ts

```
import { Component } from '@angular/core';
```

```
import { Animal } from '../models/animal';
```

```
@Component({
  selector: 'app-animal',
  templateUrl: './animal.component.html',
  styleUrls: ['./animal.component.css']
})

export class AnimalComponent {
  // 1. Definimos el array de animales

  animales: Animal[] = [
    { id_entidad: '001', nombre: 'Leo', especie: 'León', fecha_nacimiento: new Date(),
      email_cuidador: 'ana@zoo.com', fecha_registro: new Date(), clave: '01' },
    { id_entidad: '002', nombre: 'Kira', especie: 'Tigre', fecha_nacimiento: new Date(),
      email_cuidador: 'paco@zoo.com', fecha_registro: new Date(), clave: '02' }
  ];

  // 2. Funciones obligatorias (pero vacías por ahora)

  actualizar(): void {
    // El botón no tiene que hacer nada según requisitos
  }

  eliminar(): void {
    // El botón no tiene que hacer nada según requisitos
  }
}
```

1. Hoja de Trucos: Comandos de Consola

Estos son los comandos que usarás para levantar todo desde cero:

- **Crear el proyecto:** `ng new nombre-app` (selecciona CSS/Tailwind y sin SSR).
 - **Crear componente:** `ng g c componentes/nombre-componente`
 - **Crear interfaz (modelo):** `ng g i models/nombre-interfaz`
 - **Levantar el servidor:** `ng serve`
-

2. Estructura de Archivos Clave

Asegúrate de que tus archivos estén organizados así para no perderte:

- `src/app/models/`: Tus interfaces (el "contrato" de los datos).
 - `src/app/componentes/`: Tus componentes (lógica, HTML y CSS).
 - `src/app/app.config.ts`: Configuración global (rutas, HTTP, hidratación).
 - `src/app/app.routes.ts`: Definición de las URLs de la app.
-

3. Checklist de Errores Comunes

Antes de decir "he terminado", comprueba estos 5 puntos. Suelen ser los que más fallan en los exámenes:

1. **¿Importaste CommonModule?** Si usas `*ngFor` o `*ngIf` en un componente Standalone, este import es obligatorio en el archivo `.ts`.
 2. **¿Importaste FormsModule?** Si usas `[(ngModel)]` para capturar datos de un input, necesitas este import en el `.ts`.
 3. **¿El selector está en el app.component.html?** Si no pones `<app-animal></app-animal>` (o el nombre de tu componente) en el HTML principal, no verás nada en el navegador.
 4. **¿Los datos coinciden con la Interfaz?** Asegúrate de que los objetos de tu array tengan exactamente las mismas propiedades que definiste en el modelo (ojo con las mayúsculas).
 5. **¿El CDN de Tailwind está puesto?** Si las clases de diseño no funcionan, añade el script de Tailwind en el `<head>` del `index.html`.
-

4. Recordatorio de Sintaxis Rápida

- **Pintar variable:** `{{ variable }}`

- **Bucle:** *ngFor="let item of lista"
- **Condición:** *ngIf="lista.length === 0"
- **Evento de clic:** (click)="miFuncion()"
- **Formulario:** [(ngModel)]="objeto.propiedad"
- **Transformar fecha:** {{ fecha | date:'dd/MM/yyyy' }}