

Clases con clases

En este capítulo, vamos a aumentar la complejidad para entender todas las posibilidades que nos permiten las clases, con la ayuda de TypeScript.

La clase *Banda* que creaste probablemente se ve así:

```
1 class Banda {
2     members: string[];
3     albums: object[];
4     constructor(members: string[], albums: object[]) {
5         this.members = members;
6         this.albums = albums;
7     }
8 }
```

En el desafío, le indicamos al *constructor* que *albums* tiene que recibir el tipo de dato *object[]*, o sea, cualquier array de objetos. Esa es la única condición.

Ahora, imaginemos que queremos ser más específicos. En lugar de indicarle a *albums* que tiene que recibir un *object[]*, podemos decirle que espere recibir otra clase, a la que le vamos a definir ciertas características.

```
1 class Album {
2     title: string;
3     songs: string[];
4 }
5 class Banda {
6     members: string[];
7     albums: object[];
8     constructor(members: string[], albums: Album[]) {
9         this.members = members;
10        this.albums = albums;
11    }
12 }
```

Entonces, ahora especificamos en la clase *Banda* que uno de los parámetros de su *constructor* va a ser otra clase, a la que llamamos *Albums*. A su vez, le definimos a la clase *Albums* dos propiedades: *title*, que espera un *string*, y *songs*, que espera un *array* de *strings*.

De esta forma, podemos ir formando una especie de red de clases, en la que unas nos ayudan a definir otras. El objetivo es que las estructuras de las clases sean análogas a la relación que tienen los objetos en el mundo real, tal como se espera en el paradigma de la programación orientada a objetos.