

En nuestra aplicación de películas queremos poder buscar por varios atributos a la vez. Por ejemplo, películas de acción que duren menos de 2 horas. ¿Cómo podemos hacer esto?

En **MovieService** vamos a implementar un método como el siguiente:

```
public Collection<Movie> findMoviesByTemplate(Movie template) { ... }
```

En este caso, la película “template” que pasamos como argumento será una plantilla que puede tener varios atributos rellenos, para indicar por cuáles queremos buscar. Los atributos que tengan valor null no se usarán para la búsqueda.

Nota: Nos irá bien cambiar el tipo del atributo **minutes** a **Integer** en lugar de **int**, porque así podremos indicar un valor **null**. En Java, los tipos primitivos como **int**, **boolean** o **double** no pueden tener valor *\*null*.

En un test podríamos buscar películas de acción de menos de 2h 30m de la siguiente manera:

```
String name = null; // no queremos buscar por nombre
```

```
Integer minutes = 150; // 2h 30m
```

```
Genre genre = Genre.ACTION;
```

```
Movie template = new Movie(name, minutes, genre);
```

```
Collection<Movie> movies =
```

```
    movieService.findMoviesByTemplate(template);
```

```
assertThat(getMoviels(movies), is(Arrays.asList(7)) );
```

Nota: El código anterior se podría reducir (haciendo [inline](#) de las variables). En los tests es conveniente hacer esto, pues suele quedar más claro. El código compactado quedaría así:

```
assertThat(
    getMoviels(movieService.findMoviesByTemplate(
        new Movie(null, 150, Genre.ACTION))),
    is(Arrays.asList(.....))
);
```

Piensa diferentes combinaciones de búsquedas y escribe tests para ellas. Por ejemplo, prueba a buscar por nombre y duración; por nombre y género; por nombre, género y duración; etc. Si añadiste el campo director (reto anterior), también prueba a buscar por director junto a otros atributos.

Si en la búsqueda indicamos el id de la película, se ignorarán los demás atributos, ya que al buscar por id no es necesario indicar ningún atributo más.

Si los minutos indicados son negativos, la función lanzará una **IllegalArgumentException**.