Manejo básico de errores

Desarrollo de un API REST con Spring Boot

Posibles situaciones de error

- Al usar nuestra api podemos encontrarnos con posibles situaciones de error.
- Algunas pueden venir asociadas a la lógica de negocio.
- Otras son comunes
    - Buscar un producto por su ID y que dicho producto no exista.

Hasta ahora

- Tratamiento “artesanal”
- No hemos utilizado el mecanismo de ~~errores~~ excepciones de Java.
- Podemos crear excepciones y *aderezarlas un poco*.

Excepción

- Producto no encontrado (peticiones GET, PUT y DELETE)

```
public class ProductoNotFoundException extends RuntimeException {
	public ProductoNotFoundException(Long id) {
		super("No se puede encontrar el producto con la ID: "
+ id);
	}
}
```

Respuesta en el cliente

- Si lanzamos las excepción tal cual, el cliente recibe un error 500
- Sin embargo, nos interesaría más que fuera 404.
- Spring tiene la solución.

@ResponseStatus

- Nos permite indicar un código de estado y un texto de respuesta.
- Anotando la excepción

```
@ResponseStatus(HttpStatus.NOT_FOUND)
public class ProductoNotFoundException extends RuntimeException {...}
```

Modificaciones en nuestro ejemplo

- En lugar de manejar los errores con if/else y devolver constantemente ResponseEntity<?>, podemos simplificar nuestro código.

Reto

- Crear algunos errores más. Por ejemplo
  - Uno que se lance cuando no haya productos en el catálogo y se soliciten todos.
  - Uno que se lance cuando no se haga una petición correcta de creación de un nuevo Producto.
  - ...