

Back End

**Carrera
Programador
full-stack**

BAJAS

Pensemos

- ¿Qué verbo se usaba en una API REST para borrar un recurso?
- ¿A qué URL responde?
- ¿Qué debería hacer ese end point?
- ¿Dónde se programa cada parte?

Controller

```
@Delete('/:id')
deleteTrackById(@Param('id') id: number) {
  return this.trackService.deleteTrackById(id);
}
```

Service

```
async deleteTrackById(id: number): Promise<any> {
  const res = await fetch(BASE_URL + id, {
    method: 'DELETE',
  });
  const parsed = await res.json();
  return parsed;
}
```

Back End

**Carrera
Programador
full-stack**

MODIFICACIONES

Lógica

- Similar al **POST**
- En lugar de **crear** una pista, vamos a **modificar** una que ya existe.
- Vamos a necesitar el identificador de la pista a modificar (url params).
- También los nuevos datos de la pista

¿Cuál es la URL que vamos a crear?

Combina **parámetros** y el acceso al **body** de la request

Controller

TS track.controller.ts M X

src > track > TS track.controller.ts > ...

```
32 | @Put('/:id')
33 | updateTrackById(@Param('id') id: number, @Body() body): Promise<void> {
34 |     return this.trackService.updateTrackById(id, body);
35 | }
36 | }
```

Service

TS track.service.ts X

src > track > TS track.service.ts > TrackService > createTrack

```
43   async updateTrackById(id: number, body: Track): Promise<void> {
44       const isTrack = await this.getTrackById(id);
45       if (!Object.keys(isTrack).length) return; //early return
46       const updatedTrack = { ...body, id };
47       console.log('updatedTrack', updatedTrack);
48       const res = await fetch(BASE_URL + id, {
49           method: 'PUT',
50           headers: {
51               'Content-Type': 'application/json',
52           },
53           body: JSON.stringify(updatedTrack),
54       });
55       const parsed = await res.json();
56       return parsed;
57   }
58 }
```

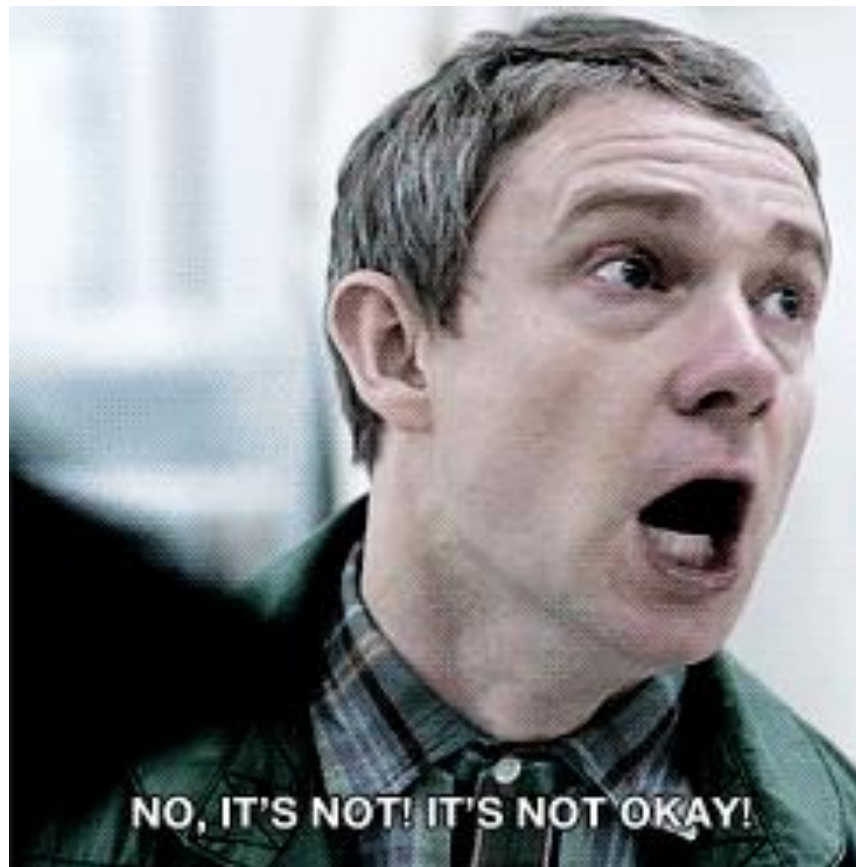
¡ Está todo OK !

Hemos terminado de programar los endpoints de nuestra primera API. Los usuarios pueden hacer peticiones de tipo POST, GET, PUT y DELETE (Operaciones CRUD).



¿ Está todo OK ?

Y los códigos de respuesta estándar HTTP? Y las validaciones? Y los DTO? Y Candela? Y la moto?



Ahora nos resta convertir nuestra API básica en una que contemple estos temas, ineludibles en el desarrollo de software profesional.