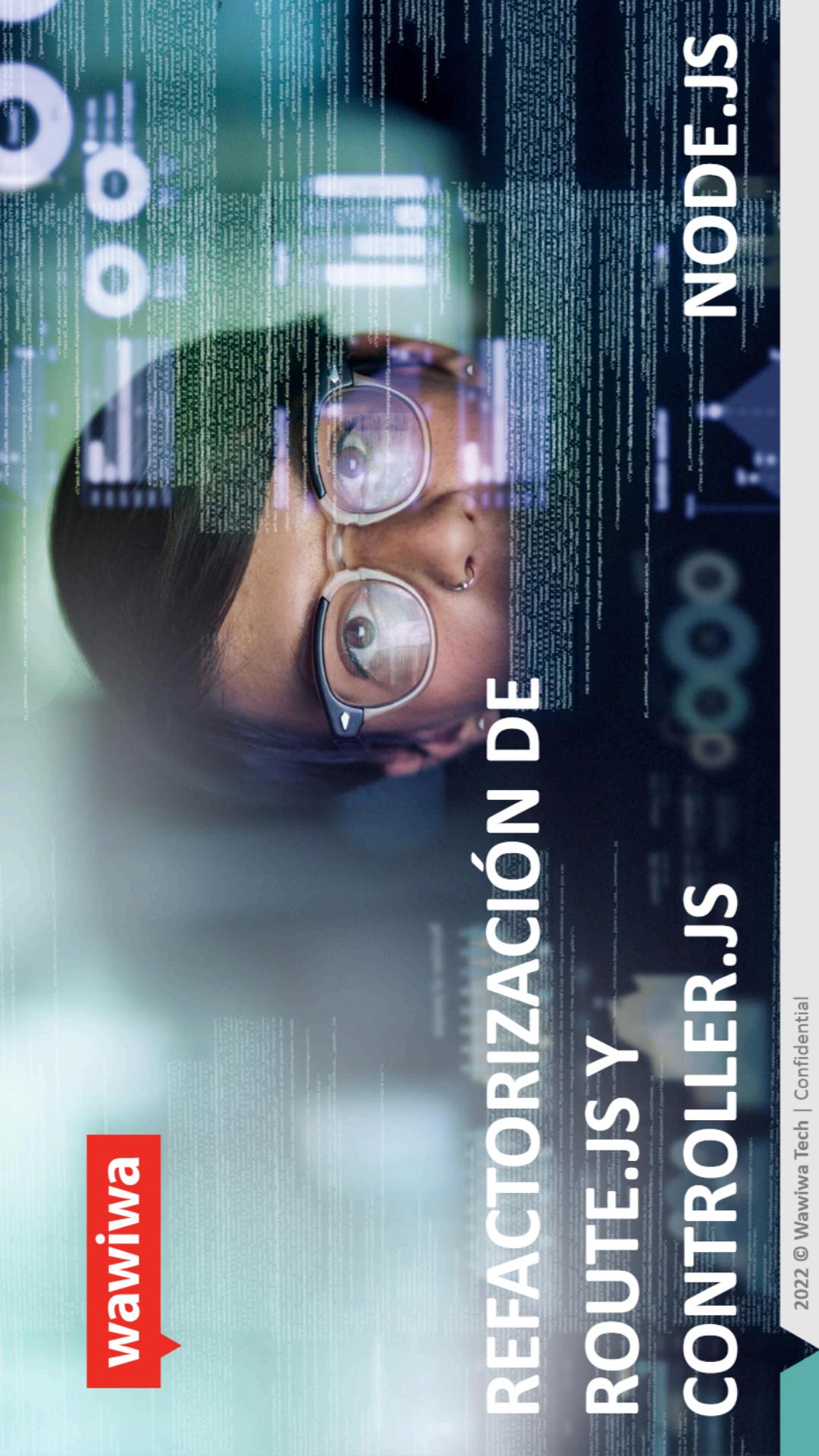




wawiwa

# REFACTORIZACIÓN DE ROUTE.JS Y CONTROLLER.JS

# NODE.JS



# Refactorización

Refactorización de código es definido como el proceso de reestructurar código de computadora sin cambiar o agregar a su comportamiento externo y funcionalidad. Existen muchas maneras de refactorizar, pero la mayoría de las veces consiste en aplicar una serie de métodos estandarizados.

# Crea carpetas

✓ part6-middle

✓ api

✓ toyController.js

✓ toyRoute.js

> assets / toys

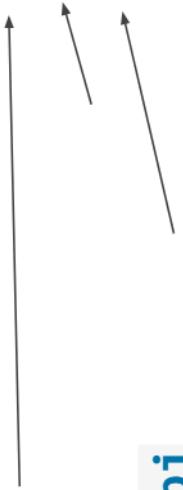
> data

> node\_modules

{} package-lock.json

{} package.json

✓ server.js



1. Crea la carpeta api
- a. Crea el archivo toyController.js
- b. Crea el archivo toyRoute.js

## toyRouter.js

part6-middle/api/toyController.js

```
const express = require('express');

const toyRouter = express.Router();

const toyController = require('./toyController');

toyRouter.get('/', toyController.getAllToys);

toyRouter.post('/toys', toyController.createToy);

toyRouter.get('/:id', toyController.getToyById);

toyRouter.patch('/:id', toyController.updateToyById);

toyRouter.delete('/:id', toyController.deleteToyById);

module.exports = toyRouter;
```

# toyRouter.js

```
chap11-refactor > part6-middle > api > toyController.js > ...
  1 const fs = require('fs')
  2 let toys = JSON.parse(fs.readFileSync("./data/data.json", 'utf-8'))
  3 > exports.getAllToys=(req, res)=>{ ...
  9   }
10 > exports.createToy=(req, res)=>{ ...
22   }
23 > exports.getToyById=(req, res)=>{ ...
36   }
37 > exports.updateToyById=(req, res)=>{ ...
59   }
60 > exports.deleteToyById=(req, res)=>{ ...
78   }
79 }
```

Exporta todas  
las funciones



# Server.js

```
const express = require("express");  
  
const app = express();  
  
const toyRouter = require("./api/toyRoute");  
app.use('/api/v1/toys', toyRouter);  
app.use(express.json());  
  
const port = 3000;  
  
app.use('/assets', express.static(__dirname + '/public'));  
app.listen(port);
```

require **toyRoute.js**

use **toyRouter** when user  
request **/api/v1/toys**

Use **express.json** for  
body content

## Ejercicio de refactorización 3 capítulo 3 - a router.js y controller.js

Solicitud	Respuesta	Método	Comentario
/api/v1/players	Todos los "players"	Get	Obtener todos los "players" de players.json
/api/v1/players	El "player" creado"	Post	Crea un nuevo Player
/api/v1/players/:id	"Player" por su id	Get	Obtener el "player" por su id
/api/v1/players/:id	Actualizar "player"	Patch	Actualizar "player" por su id
/api/v1/players/:id	Borrar "player"	Delete	Borrar "player" por su id



wawiwa

¿Preguntas?