Enrutadores en Express.js - Explicación para Clase

Explicación con Analogía y Ejemplo Real

¿Qué es un enrutador en Express?

Una analogía para empezar...

Imaginá que estás construyendo un edificio de oficinas. Cada oficina tiene una función diferente: una es de recursos humanos, otra de ventas, otra de atención al cliente.

En este edificio, hay una recepción principal (como tu archivo index.js) que se encarga de derivar a cada persona que entra hacia la oficina correcta, según el motivo de su visita.

Los enrutadores en Express.js funcionan exactamente igual.

El enrutador es como una oficina especializada dentro de tu aplicación. Tiene su propia lógica, su propio equipo, y se ocupa exclusivamente de un tipo de tarea: usuarios, productos, tareas, etc.

¿Por qué usar enrutadores?

A medida que tu aplicación crece, tener todas las rutas en un solo archivo se vuelve un caos. Separar por enrutadores permite:

Mantener tu código organizado

Facilitar el mantenimiento y escalabilidad

Hacer más fácil trabajar en equipo

¿Y qué hace exactamente un enrutador?

Enrutadores en Express.js - Explicación para Clase

Un enrutador en Express es una mini-aplicación que maneja un grupo de rutas relacionadas.

Por ejemplo, el enrutador de usuarios puede tener:

router.get("/usuarios", ...)

router.post("/usuarios", ...)

router.delete("/usuarios/:id", ...)

¿Cómo se conecta todo? (Basado en tu ejemplo)

1. Tu index.js es la puerta de entrada principal

En index.js, estás leyendo todos los archivos que hay en la carpeta routes/:

const routeFiles = fs.readdirSync("./src/routes");

Después, los importás dinámicamente y los montás con una ruta base común:

app.use("/api/v1", route.default);

Eso significa que si uno de tus archivos de rutas define esto:

router.get("/usuarios", ...)

Entonces, la ruta completa en tu API será:

/api/v1/usuarios

Tu index.js no sabe ni le importa qué hace cada archivo. Solo se encarga de "derivar" a la persona a la

oficina correcta. Eso es modularidad y responsabilidad única.

¿Y qué hay dentro de un enrutador?

Cada archivo en src/routes/ es un módulo que exporta un enrutador. Por ejemplo, el archivo usuarios.routes.js puede verse así:

```
import { Router } from "express";
const router = Router();

router.get("/usuarios", (req, res) => {
  res.json({ mensaje: "Listado de usuarios" });
});

router.post("/usuarios", (req, res) => {
  res.json({ mensaje: "Usuario creado" });
});

export default router;
```

Este archivo no necesita saber nada del servidor principal, solo se enfoca en lo suyo: manejar usuarios.

Ventajas de esta estructura modular

1. Escalable: podés seguir agregando rutas sin tocar index.js

Enrutadores en Express.js - Explicación para Clase

2. Flexible: cada enrutador puede tener su propio middleware si lo necesitás

3. Limpio: tu index.js sigue siendo claro y enfocado en configurar el servidor

Visualizando todo el flujo

index.js

app.use("/api/v1", route.default)

usuarios.routes.js /usuarios

tareas.routes.js /tareas

productos.routes.js /productos

Cliente hace GET /api/v1/tareas tareas.routes.js responde

En resumen

- Un enrutador es como una oficina especializada dentro de tu app Express
- Ayuda a separar las responsabilidades
- Te permite escalar sin perder el control
- En tu implementación, los enrutadores están en src/routes/, y el index.js los monta todos automáticamente