# Capítulo 05: Imports relativos y absolutos (alias con @/)

En este capítulo veremos cómo funcionan los imports en JavaScript, cuáles son las diferencias entre rutas relativas y absolutas, y cómo Next.js introduce el alias @/ para simplificar el código. Esto es fundamental porque los alumnos vienen de JavaScript puro (Vanilla JS), donde solo existen imports relativos.

## 1. Imports relativos en Vanilla JS

En JavaScript puro, cuando importamos archivos, debemos usar rutas relativas:  
  
Ejemplo:  
import { sumar } from "./utils/operaciones.js";  
  
Si el archivo está en otra carpeta, las rutas se vuelven más largas y difíciles de leer:  
import { restar } from "../../helpers/matematica.js";  
  
👉 Problema: las rutas con muchos '../' son poco legibles y difíciles de mantener.

## 2. Imports en Node.js + Express

En Node.js también se usan rutas relativas por defecto:  
  
const utils = require("../utils/operaciones");  
  
Para usar imports absolutos o alias, necesitamos instalar un paquete adicional como module-alias. En package.json se puede configurar:  
  
{  
 "\_moduleAliases": {  
 "@utils": "src/utils"  
 }  
}  
  
Y en el código:  
const utils = require("@utils/operaciones");  
  
👉 Conclusión: Node no soporta alias nativamente, pero se puede configurar con librerías.

## 3. Imports absolutos en Next.js

Next.js simplifica los imports con alias. Por defecto, el alias @/ apunta a la raíz del proyecto.  
  
Ejemplo:  
import Button from "@/components/Button";  
  
Esto es equivalente a:  
import Button from "../../components/Button";  
  
👉 Ventaja: el código queda más limpio y es menos propenso a errores.  
  
El alias se configura en jsconfig.json (o tsconfig.json si usamos TypeScript):  
  
{  
 "compilerOptions": {  
 "baseUrl": ".",  
 "paths": {  
 "@/\*": ["./\*"]  
 }  
 }  
}

## 4. Comparación rápida

|  |  |  |
| --- | --- | --- |
| Entorno | Soporte por defecto | Alias / Absolutos |
| Vanilla JS (navegador) | Solo relativos | No soporta alias |
| Node.js + Express | Solo relativos | Alias con librerías (ej. module-alias) |
| Next.js | Relativos y alias | Alias @/ listo para usar |

## Resumen

- En Vanilla JS solo existen imports relativos.  
- En Node.js también son relativos, pero se pueden configurar alias con librerías.  
- En Next.js existe el alias @/ por defecto, configurado en jsconfig.json.  
  
👉 Como docentes, debemos mostrar primero los imports relativos en Vanilla, luego el problema de múltiples '../', y finalmente cómo Next.js resuelve esto con el alias @/.