

# Diferenças e Usos: ESM Modules vs CommonJS

No ecossistema JavaScript, dois sistemas principais de módulos são utilizados: ESM Modules (ESM) e CommonJS (CJS). Ambos têm o mesmo objetivo — modularizar o código — mas funcionam de formas diferentes.

**Principais Diferenças:**

- **Sintaxe:** - ESM Modules: utiliza *import* e *export*. - CommonJS: utiliza *require()* e *module.exports*.
- **Carregamento:** - ESM Modules: carregamento assíncrono (suportado nativamente em navegadores modernos e Node.js). - CommonJS: carregamento síncrono (original do Node.js).
- **Escopo:** - ESM Modules: possui escopo estrito e sempre usa "strict mode". - CommonJS: não utiliza strict mode por padrão.
- **Suporte:** - ESM Modules: padrão oficial do JavaScript (ECMAScript 2015+). - CommonJS: específico do Node.js, mas amplamente usado em pacotes NPM.

**Quando usar cada um:**

- **ES Modules:** - Projetos modernos em JavaScript/TypeScript. - Aplicações front-end em navegadores. - Projetos Node.js atuais que suportam ESM (com extensão *.mjs* ou "type": "module" no package.json).
- **CommonJS:** - Projetos legados Node.js. - Ambientes que ainda não suportam ESM totalmente. - Pacotes NPM antigos escritos em CommonJS.

Em resumo, o ESM Modules é o padrão oficial e recomendado para novos projetos, enquanto o CommonJS ainda é útil para compatibilidade com código legado.