

GUÍA NODESTUDIO

Cómo hacer que Claude Code use el modelo correcto para cada tarea

Un CLAUDE.md y dos subagentes para gastar menos sin perder calidad.

⏱ 5 minutos de configuración

1 La idea

El modelo más potente es también el más caro y el más lento. Usarlo para buscar un archivo o resumir un log es pagar de más por algo que Haiku hace igual de bien. La jugada: un modelo por defecto para el trabajo normal, uno potente solo para lo difícil y uno barato para lo mecánico.

El routing

Modelo	Para qué	Cómo se usa
Fable (el potente)	Arquitectura, bugs difíciles, análisis complejo	Subagente `arquitecto`
Opus 5.5	Código, tests, refactors, documentación	Sesión principal
Haiku	Búsquedas, resúmenes, tareas simples	Subagente `rapido`

CÓMO FUNCIONA DE VERDAD

El CLAUDE.md **no cambia de modelo por sí solo**: son instrucciones, no configuración. Lo que fija el modelo es el campo `model:` de cada subagente en `.claude/agents/`. El CLAUDE.md le dice a Claude *cuándo* delegar a cada uno. La sesión principal usa el modelo que elegís con `/model`.

DOS COSAS A SABER

El subagente integrado `Explore` ya no corre en Haiku: hereda el modelo de tu sesión. Por eso conviene tener tu propio subagente en Haiku.
Según tu plan, Fable puede cobrarse con créditos de uso aparte. Claude Code te pide confirmación antes.

2 El CLAUDE.md

Pegalo en el `CLAUDE.md` de la raíz de tu proyecto (o en `~/.claude/CLAUDE.md` para todos tus proyectos).

```
## Routing de modelos

La sesión principal corre en Opus 4.5. Hacé el trabajo normal
directamente, sin delegar: escribir código, tests, refactors
y documentación.

### Delegá al subagente `arquitecto` (Fable) cuando haya:
- Decisiones de arquitectura o diseño que afectan varios módulos
- Un bug que no se resolvió al primer intento o sin causa clara
- Análisis complejo: performance, concurrencia, seguridad,
  migraciones grandes
El subagente no ve esta conversación: pasale en el pedido los
archivos, el error y lo que ya se probó.

### Delegá al subagente `rapido` (Haiku) cuando haya que:
- Buscar dónde está algo en el repo
- Resumir archivos largos, logs u output de tests
- Hacer lecturas mecánicas que terminan en un resumen corto

### No delegues cuando:
- El cambio es de pocas líneas o la respuesta es inmediata
- La tarea depende de lo que ya hablamos en esta conversación
- Explicarle la tarea al subagente lleva más que hacerla

Al terminar una delegación, decime en una línea qué subagente
usaste y por qué.
```

3 Armalo con Claude Code

No tenés que crear nada a mano. Abrí Claude Code en tu proyecto y pegale esto:

PROMPT PARA PEGAR EN CLAUDE CODE

```
Configurá routing de modelos en este proyecto:

1. Creá .claude/agents/arquitecto.md con frontmatter name: arquitecto, model: fable y una description que diga: "Arquitectura, bugs difíciles y análisis complejo. Usar solo cuando la tarea lo justifique." En el cuerpo, que investigue la causa raíz antes de proponer cambios y devuelva un diagnóstico y un plan concretos.

2. Creá .claude/agents/rapido.md con name: rapido, model: haiku, tools: Read, Grep, Glob y la description: "Búsquedas en el repo, resúmenes de archivos y logs, tareas simples de lectura. Usar proactivamente." En el cuerpo, que responda corto y con rutas de archivo.

3. Agregá al CLAUDE.md la sección "Routing de modelos" que te pego abajo.

4. Cambiá la sesión a Opus 5.5 con /model opus y avisame si hay que reiniciar para que cargue .claude/agents.

[pegá acá el CLAUDE.md de la página anterior]
```

PARA COMPROBAR QUE ANDA

Mientras corre un subagente, escribí `/tasks` : cada fila muestra en qué modelo está corriendo. Si querés forzar uno, mencionalo con `@` (por ejemplo `@arquitecto`).

4 Tres ejemplos

Tarea	Modelo	Por qué
"El checkout falla de vez en cuando y no sé por qué"	**Fable** (`arquitecto`)	Causa poco clara, posible condición de carrera: vale el costo
"Agregá validación al formulario de registro y sus tests"	**Opus 5.5** (sesión principal)	Código estándar con contexto que ya está en la conversación
"¿Dónde se llama a la API de pagos?"	**Haiku** (`rapido`)	Búsqueda mecánica: el resultado es una lista de rutas

5 Cuándo NO delegar

Cada subagente arranca de cero: no ve tu conversación, vuelve a leer el CLAUDE.md y tiene que buscar el contexto de nuevo. Eso cuesta tokens y tiempo.

> **REGLA PRÁCTICA**
> Si la tarea es cambiar dos líneas, renombrar una variable o contestar algo que ya está en la conversación, hacelo en la sesión principal. Delegá cuando el subagente se ahorre más de lo que cuesta explicarle la tarea.



NodeStudio

Automatizamos procesos de negocio con IA y sistemas a medida.

nodestudioai.com