

El CLAUDE.md de Boris Cherny.

CATEGORÍA · PRIMEROS PASOS CON CLAUDE

Boris Cherny creó Claude Code. Este es el archivo que usa para darle instrucciones permanentes a Claude sobre cómo operar para él.

QUÉ VAS A ENCONTRAR

- 01 Quién es Boris Cherny y por qué esto importa
- 02 Qué es un CLAUDE.md y cómo funciona
- 03 Su sistema de planificación y sub-agentes
- 04 Cómo Claude se auto-corrige entre sesiones
- 05 El archivo completo listo para copiar y adaptar

FICHA DEL RECURSO

Qué es, para qué sirve y cómo se implementa

FUNCIÓN

Boris Cherny creó Claude Code. Este es el archivo que usa para darle instrucciones permanentes a Claude sobre cómo operar para él.

PARA QUÉ SIRVE

Boris Cherny es el creador de Claude Code, la herramienta de AI más poderosa para escribir y gestionar código disponible hoy. La presentó en el evento de Anthropic en mayo de 2026. Como cualquier desarrollador serio, Boris tiene un sistema para operar su propia herramienta. Ese sistema vive en un archivo llamado CLAUDE.md. Es el conjunto de instrucciones permanentes que Claude lee al inicio de cada sesión. Lo que ves en este PDF es exactamente lo que Boris usa. No es una guía genérica. Es la configuración real del creador.

DÓNDE APLICARLO

Cubre: Quién es Boris Cherny y por qué esto importa; Qué es un CLAUDE.md y cómo funciona; Su sistema de planificación y sub-agentes; Cómo Claude se auto-corrige entre sesiones; El archivo completo listo para copiar y adaptar.

CÓMO IMPLEMENTARLO

- 1 Contexto: El hombre que construyó la herramienta.
- 2 Sección 1 / Workflow: Como Claude planifica y ejecuta.
- 3 Sección 1 / Workflow: Sub-agentes y auto-mejora.
- 4 Sección 1 / Workflow: Verificación, elegancia y bugs.
- 5 Task management y principios base.

CONTEXTO

El hombre que construyó la herramienta.

Boris Cherny es el creador de Claude Code, la herramienta de AI más poderosa para escribir y gestionar código disponible hoy. La presentó en el evento de Anthropic en mayo de 2026. Como cualquier desarrollador serio, Boris tiene un sistema para operar su propia herramienta. Ese sistema vive en un archivo llamado CLAUDE.md. Es el conjunto de instrucciones permanentes que Claude lee al inicio de cada sesión. Lo que ves en este PDF es exactamente lo que Boris usa. No es una guía genérica. Es la configuración real del creador.

QUE ES UN CLAUDE.MD

Un archivo de texto en la raíz de tu proyecto.

Claude lo lee al inicio de cada sesión.

Define como planifica, como se corrige,
cuando pedir permiso y cuando ejecutar solo.

Sin este archivo: Claude empieza de cero cada vez.

Con este archivo: Claude ya sabe como trabajar contigo.

SECCION 1 / WORKFLOW

Como Claude planifica y ejecuta.

La primera sección del archivo define el flujo de trabajo. Boris divide esto en seis reglas concretas que Claude sigue en cada tarea.

1. PLAN NODE DEFAULT

Enter plan mode for ANY non-trivial task

(3+ steps or architectural decisions)

If something goes sideways, STOP and re-plan.

Don't keep pushing.

Use plan mode for verification steps,
not just building.

Write detailed specs upfront to reduce ambiguity.

SECCION 1 / WORKFLOW

Sub-agentes y auto-mejora.

Las siguientes tres reglas cubren cómo Claude distribuye trabajo entre sub-agentes y cómo aprende de sus propios errores entre sesiones.

2. SUBAGENT STRATEGY + 3. SELF-IMPROVEMENT LOOP

Use subagents liberally to keep main context clean.
Offload research, exploration and parallel analysis.
One task per subagent for focused execution.
After ANY correction from the user:
update tasks/lessons.md with the pattern.
Write rules that prevent the same mistake.
Review lessons at session start for relevant project.

SECCION 1 / WORKFLOW

Verificacion, elegancia y bugs.

Las tres reglas finales del workflow cubren cómo Claude verifica que el trabajo realmente funciona, cuándo buscar la solución más limpia y cómo manejar bugs sin pedir ayuda.

4. VERIFICATION / 5. ELEGANCE / 6. BUG FIXING

Never mark a task complete without proving it works.
Ask: 'Would a staff engineer approve this?'
For non-trivial changes: pause and ask
'is there a more elegant way?'
Skip this for simple, obvious fixes.
When given a bug report: just fix it.
Don't ask for hand-holding.
Zero context switching required from the user.

SECCION 2 Y 3 / TAREAS Y PRINCIPIOS

Task management y principios base.

La segunda seccion cubre cómo Claude gestiona tareas activas. La tercera define los principios que aplica a cualquier decision técnica.

TASK MANAGEMENT + CORE PRINCIPLES

Plan First: Write plan to tasks/todo.md
with checkable items.
Verify Plan: Check in before starting implementation.
Track Progress: Mark items complete as you go.
Capture Lessons: Update tasks/lessons.md
after corrections.
Simplicity First: Make every change as simple
as possible. Impact minimal code.
No Laziness: Find root causes. No temporary fixes.
Minimal Impact: Only touch what's necessary.

SIGUIENTE PASO

Ahora úsalo tú. Adapta cada regla.

Este archivo es el punto de partida. Las reglas de Boris están diseñadas para desarrollo de software, pero la estructura funciona para cualquier tipo de trabajo con Claude. Cada semana mostramos cómo usar estas Sin tecnicismos. Solo lo que funciona.

MÁS GUÍAS GRATIS EN [377CREATIVE.COM/RECURSOS](https://377creative.com/recursos)