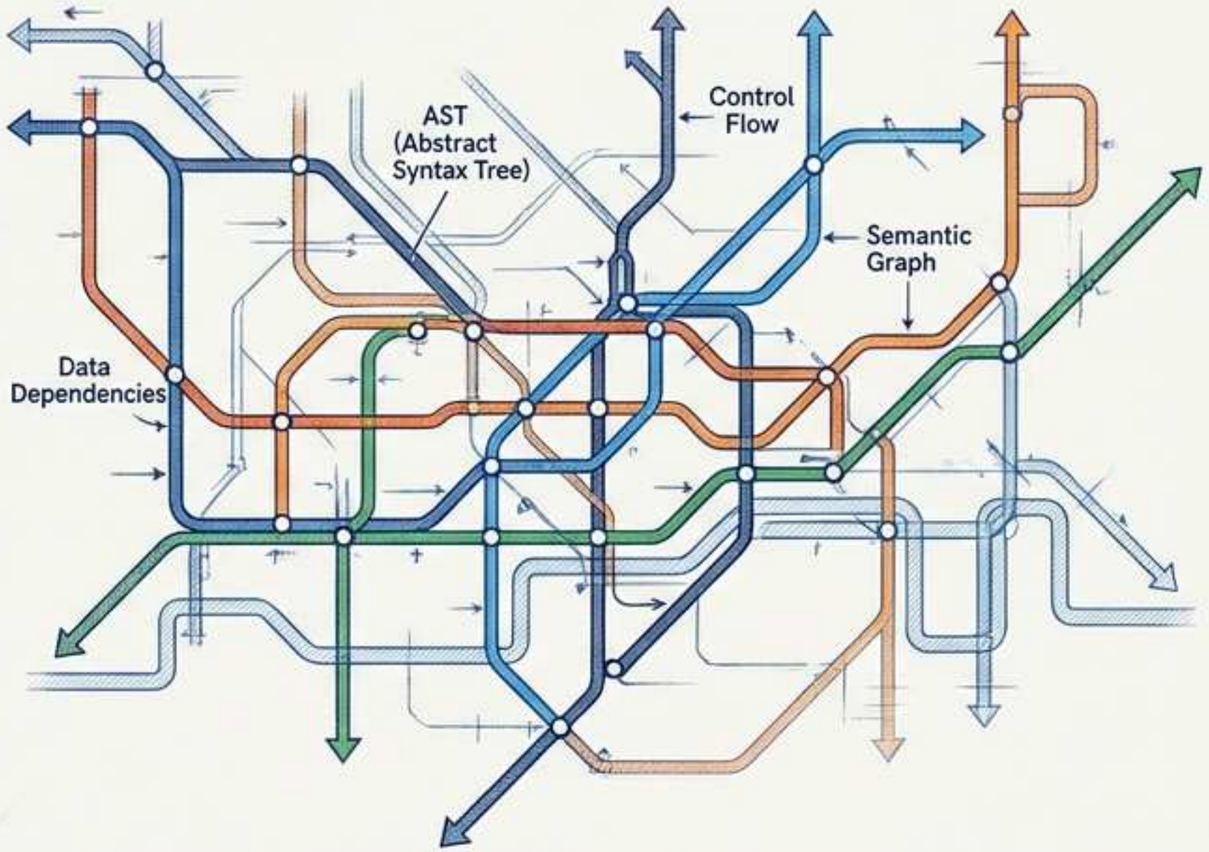
Graphify: Graphify: El Plano del Metro de tu Código

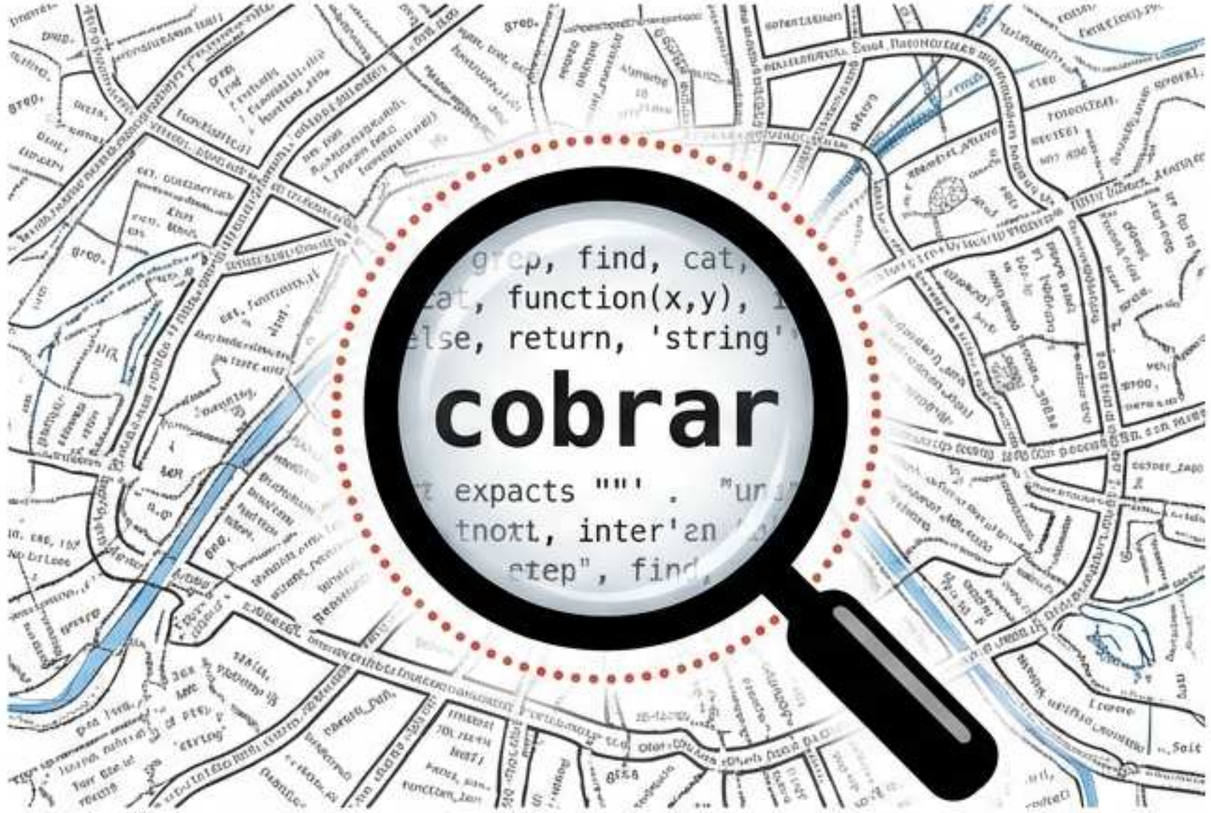
Cambiando tokens por precisión estructural en agentes de IA.




```
> graphify extract . --code-only
[OK]
```

Gemini Notebook

El Agente Ciego (Grep / Búsqueda Textual)

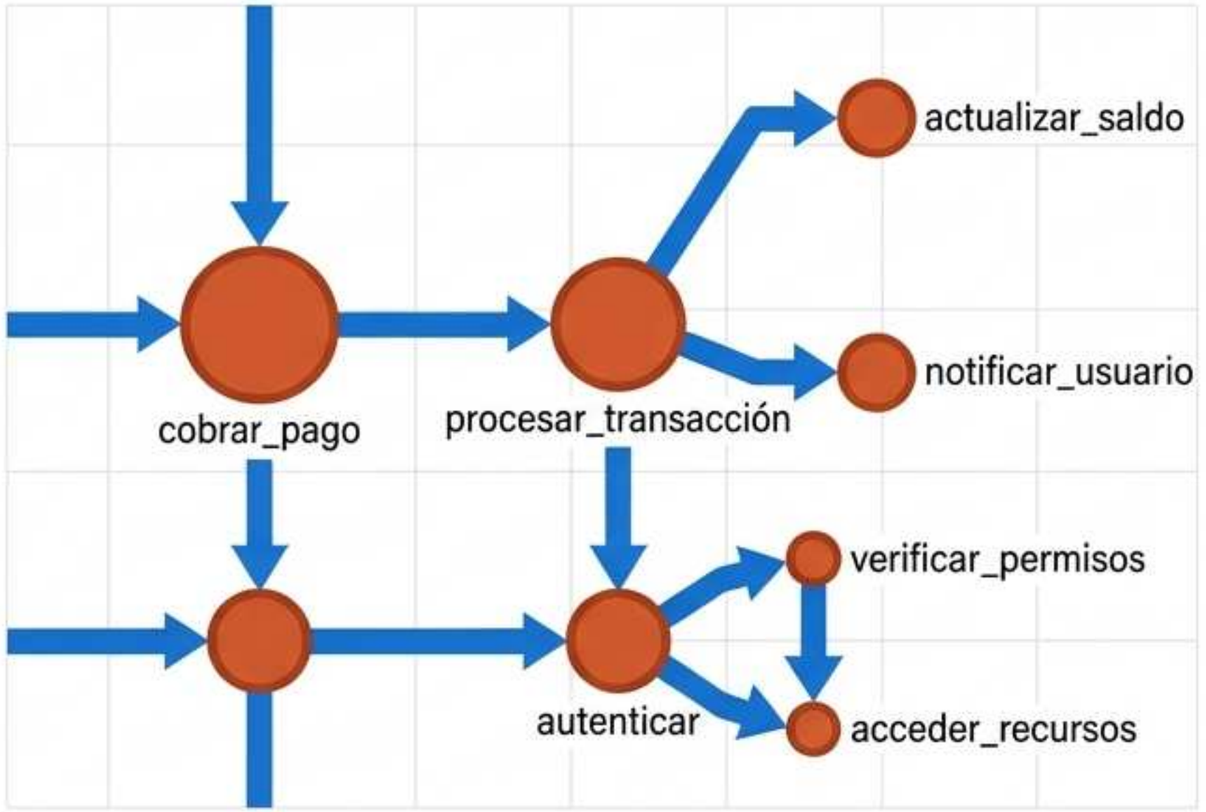


Devuelve líneas de texto aisladas. Se le escapan llamadas con nombres distintos o despachos dinámicos.

Métrica

Un grep en `datasette/app.py` obliga a leer 146 KB de contexto.

El Agente Estructural (Grafo del Código)



Guarda símbolos y dependencias direccionales (quién llama a qué).

Métrica

Una consulta topológica devuelve solo el entorno relevante (ej. 257 bytes).

Gemini Notebook

El Experimento Datasette: Mito vs. Realidad

84 sesiones de Claude Code sobre el repositorio real (34k líneas de app, 54k de tests).

+ 13%

Más tokens consumidos (mediana por sesión).

Mito destruido: Graphify en repositorios medianos no ahorra tokens.

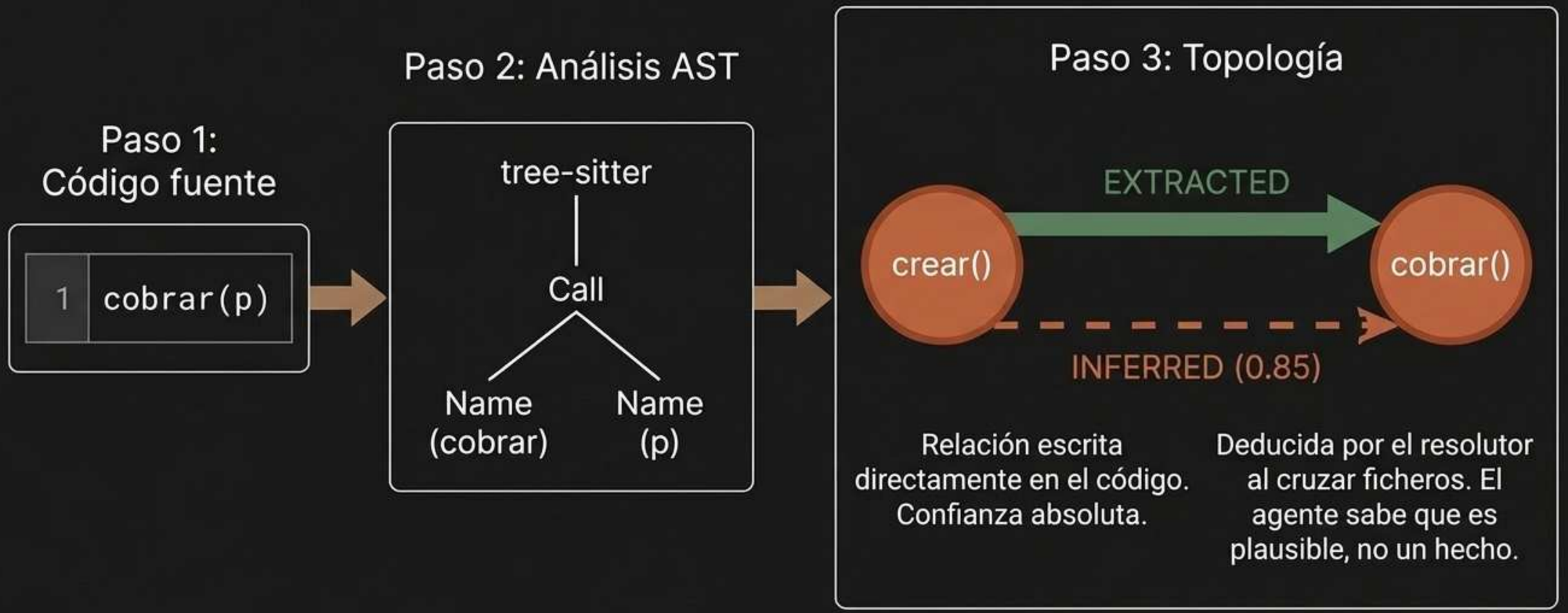
96%

Precisión en preguntas de arquitectura (frente a un 72% sin grafo).

Métrica	Sin Grafo	Con Grafo
Preguntas de Impacto (¿Qué rompe esto?)	100%	100%
Cadena de llamadas completa	92%	100%

Gemini Notebook

La Anatomía de una Arista



Gemini Notebook

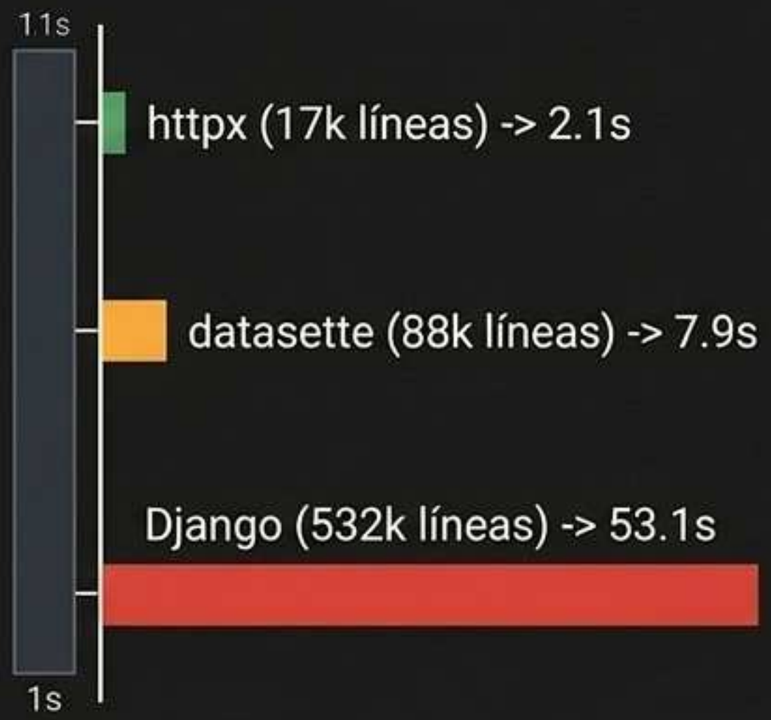
El Embudo de Costos: Tres Pasadas



Gemini Notebook

Rendimiento en Frío y el Ruido del Frontend

Velocidad tree-sitter



El análisis estático no bloquea el flujo de trabajo.

El Blindaje del .graphifyignore



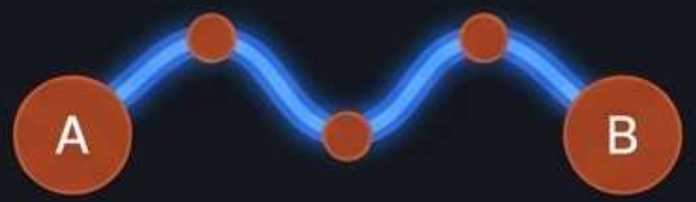
En el primer escaneo, 1,537 nodos (20% del grafo) provenían de un solo fichero: **cm-editor-6.0.1.bundle.js**.

```
datasette/static/cm-editor-*.bundle.js
```

Añadirlo a .graphifyignore reduce el tiempo a 5.8s y elimina el ruido.

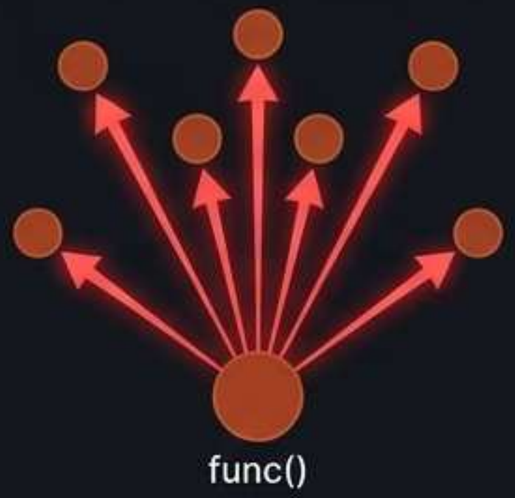
Gemini Notebook

Interrogando la Topología




```
graphify path 'A' 'B'
```

La Ruta Segura: Descubrir cómo interactúan dos subsistemas alejados.




```
graphify affected 'func()'
--depth 1
```

El Radio de Explosión: Saber exactamente qué se rompe antes de cambiar una firma.




```
graphify query --seed
'node' --budget 800
```

El Escáner: Explorar el entorno sin devorar la ventana de contexto.

Gemini Notebook

Matriz de Paradigmas de Búsqueda

Dimensión	Búsqueda Vectorial (Semántica)	Servidor LSP (IDE)	Grafo del Código (Estructural)
¿En qué es excelente?	Encontrar conceptos o intenciones por similitud de texto.	Precisión exacta mientras editas un solo lenguaje.	Revelar conexiones remotas y arquitectura entre múltiples ficheros.
Responde a ¿qué se rompe si toco esto?	❌ No, solo encuentra textos parecidos.	⚠️ En parte, pero consulta a consulta.	✅ Sí, trazando las aristas dependientes (affected).
Costo de Contexto	Alto (inserta bloques enteros de texto).	Bajo.	Medio (devuelve sub-grafos filtrados).

Gemini Notebook

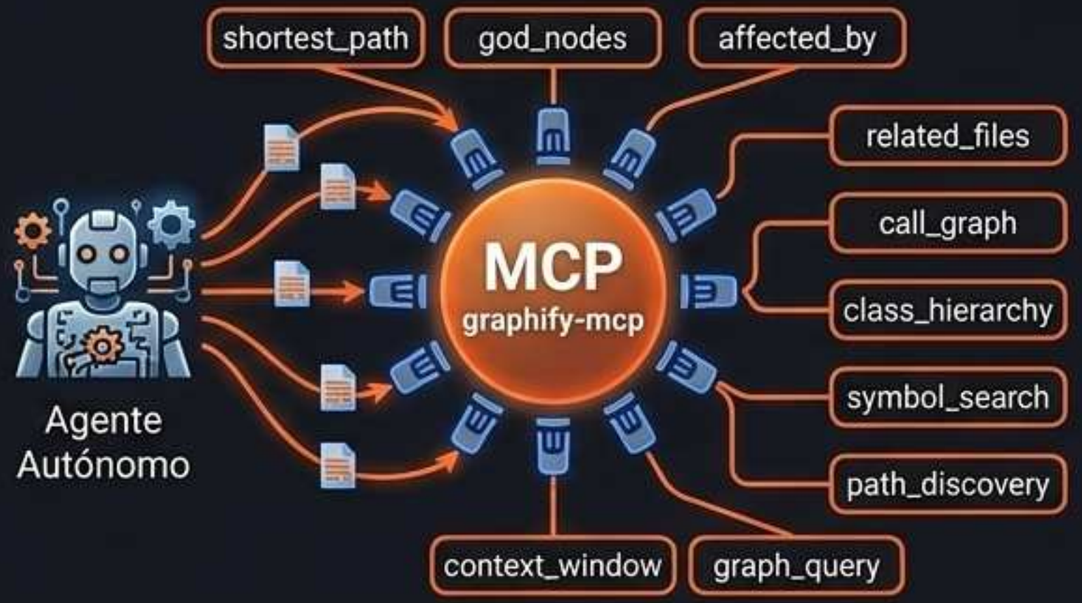
Inyectando el Mapa en el Agente

Claude Code: Hooks & Control



Modo Estricto (--strict). Si el agente intenta leer un fichero a ciegas, el hook lo deniega y le obliga a consultar el grafo primero, inyectando los símbolos cercanos en la advertencia.

OpenCode: Servidor MCP



Servidor MCP (graphify-mcp). Expone 10 herramientas topológicas nativamente. No bloquea al agente, pero este usa las herramientas por iniciativa propia para planificar.

Gemini Notebook

El Mapa en Acción (Datasette Issue 2994)

Un bug en search_filters() que lanza un error 500.

1. Orientación topológica



En lugar de hacer grep en 200 ficheros, el agente ejecutó query 'SearchFilters' y path 'search_filters()' 'Filters'.

2. Descubrimiento del contexto



Descubrió que **search_filters()** (88 líneas) es llamada por _table_filters(), conectada directamente a TableResource.

3. Ejecución quirúrgica

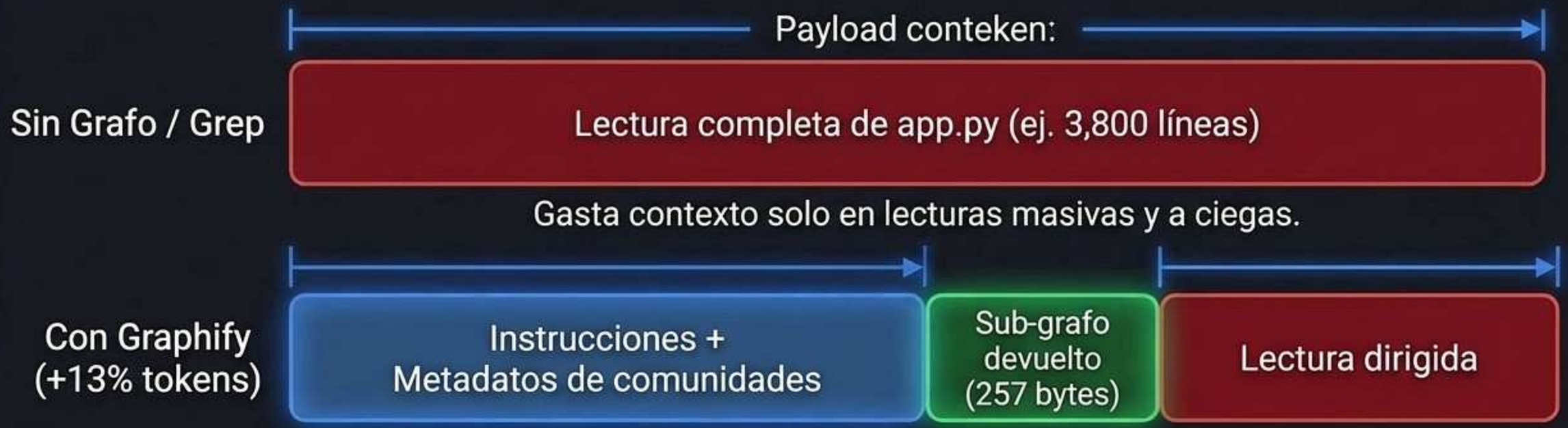
Sabiendo exactamente dónde afectaría el cambio, propuso el parche directo:

```
- search_col = key.split('_search_', 1)[1]
+ if not key.startswith('_search_'):
+     raise BadRequest('Invalid search parameter')
+ search_col = key[len('_search_'):]
```

El grafo no escribió el fix; construyó el mapa para que el agente pusiera el código y los tests en el lugar exacto.

Gemini Notebook

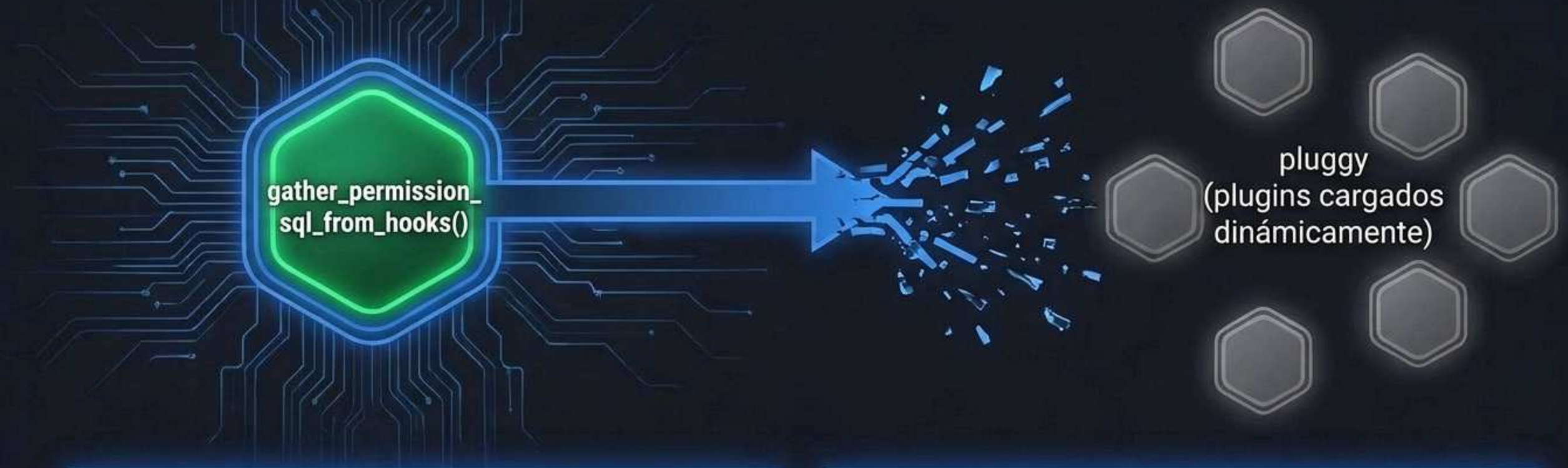
El Precio de la Estructura: ¿Por qué gasta más tokens?



El grafo le dice al agente ***dónde*** mirar, no le evita mirar.
Añade una fase de orientación topológica a la sesión.
Estás pagando un 13% de sobrecarga de contexto para evitar que el agente alucine la arquitectura.

Gemini Notebook

El Punto Ciego: Lo que el Grafo no puede ver



El Despacho Dinámico es Invisible: El análisis estático sabe que el gancho (el enchufe) existe, pero es ciego a quién lo implementa en tiempo de ejecución.

Ruido en Monorepos: En proyectos gigantescos, los nodos centrales suelen ser utilidades de pruebas genéricas (TestCase), no la arquitectura real. Requiere un riguroso .graphifyignore.

Gemini Notebook

Matriz de Adopción Pragmática

Escenario	Acción	Razón
Repositorio Pequeño / Bien Conocido	No usar.	Un simple grep es suficiente. El grafo solo añadirá latencia y consumirá más contexto del necesario.
Repositorio Mediano y Desconocido	Adoptar Local.	Generar grafo con –code-only (cero filtración de datos). Vital para mapear arquitectura rápidamente.
Código Legacy o Refactorización Profunda	Mandatorio.	Ejecutar graphify affected siempre antes de alterar una firma central. Usar modo –strict o servidor MCP.

“Si tu agente ya acierta buscando texto, un grafo solo lo hará más lento. Si rompe la arquitectura, el grafo es la brújula que le faltaba.”

Gemini Notebook