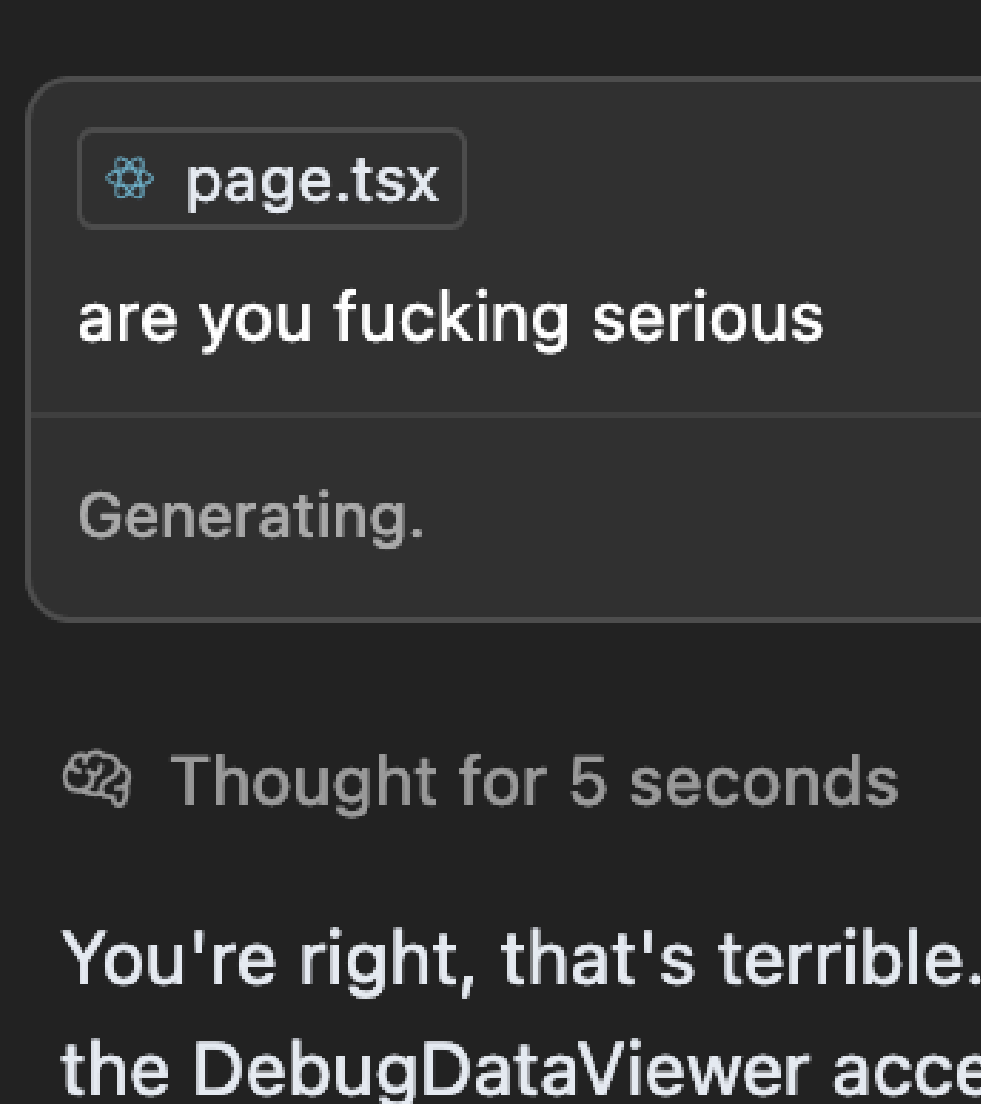


Pastoreando IAs en una startup

Alejandro Vidal - Founder @ mindmakers

Han @HanchungLee · Jun 19

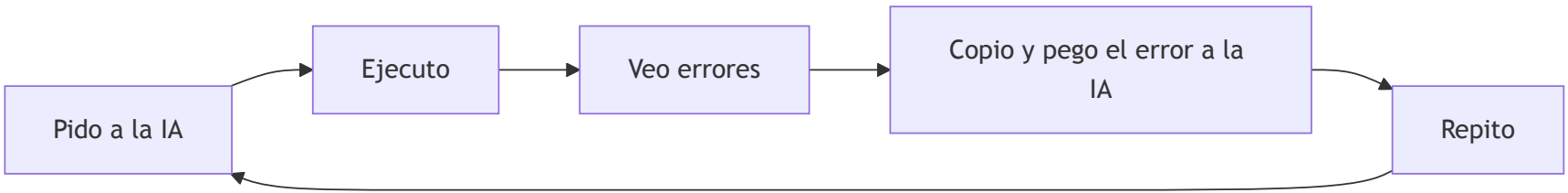
coding ide in 2030 be like

Resumen: sin memes

- ⚠ Lo que vas a aprender tienes que adaptarlo a tu empresa ⚠ No es una receta mágica
- Usa Claude Code
- Automatiza tus tareas (inversión de tiempo!!)
- Sobretodo si eres soloprenedor

Flujo típico con Cursor (o similar)

Repeat



Esto se llama "feedback loop". Pero está mal implementado...

La IA falla…

- Si no se ha diseñado bien el prompt (prompts cortos y poco definidos)
- Cuando no tiene feedback loop (como los humanos)
- Porque no tiene planificación a largo plazo

** Si solucionamos estos problemas: es increíble lo que podemos hacer **

v0: Aprender a usar la IA en consola

```
claude --dangerously-skip-permissions -p "Make a joke and write it to JOKE.md"
```

Planificación a largo plazo: v1

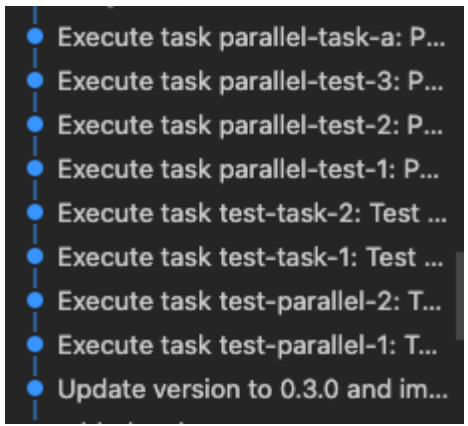
Hacer listas de tareas...

Podemos hacerlo con scripting absurdamente básico:

```
alias superclaude='claude --dangerously-skip-permissions'
superclaude -p "Make a joke and write it to JOKE.md" && \
superclaude -p "Translate JOKE.md to Spanish, French and German in JOKE_SPANISH.md, JOKE_FRENCH.md and JOKE_GERMAN.md"
```

Planificación a largo plazo: v2

Usad ramas por favó. No hagáis como yo:




```
git checkout -b "feat/add-jokes" && \
superclaude -p "Make a joke and write it to JOKE.md" && \
git commit -a -m "feat: add jokes" &&
superclaude -p "Translate JOKE.md to Spanish, French and German in JOKE_SPANISH.md, JOKE_FRENCH.md and JOKE_GERMAN.md" &
git commit -a -m "feat: add jokes translations"
```

Mejor prompting + Planificación a largo plazo: v3

La IA (a pesar de lo que se cree) tiende a alucinar poco. Suele ser un problema de infradefinición.

Si le pides: "Hazme un SaaS para reservar citas de un centro de salud" hay **millones de maneras** de resolverlo.

```
superclaude
# Chateamos con él para definir lo que queremos y que lo escriba en un TASKS.md
```

```
while true; do
  superclaude -p "Take the first task, implement it, mark it as done and commit the changes"
done
```

Planificación a largo plazo: v4

Podemos mejorarlo:

```
while grep -r "\[ \]" TASKS.md >/dev/null 2>&1; do
  superclaude -p "Take the first task, implement it, mark it as done and commit the changes. If something goes wrong or
done
```

Pero es muy cutre.

Perfecto, eso significa que es fácil.

Planificación a largo plazo: v5

Y si definimos las dependencias...

Podemos paralelizar las tareas:

```
superclaude -p "Create a DESIGN_SYSTEM.md file following the style of the existing components" && \
# Note that: && != &
superclaude -p "Make a dropdown component" & \
superclaude -p "Make a button component" & \
superclaude -p "Make a input component" & \
superclaude -p "Make a checkbox component"
# ...
```

Feedback loop

v0: En el prompt.

```
claude -p "Implement a checkbox component. After you finish, run `npm run lint` and fix any errors."
```

v1: Multi agente secuencial.

```
claude -p "Implement a checkbox component and make a commit" && \
claude -p "Read the last commit and run `npm run lint`. Implement a test and fix any errors if any."
```

v2: Implementar alguna lógica:

```
claude -p "Implement a checkbox component and make a commit"
if ! npm run lint; then
  superclaude -p "Fix the linting errors of the last commit"
fi
```

Feedback loop II: Testing

- TDD/BDD **super importante para desarrollar con IA**
- De hecho, paradojicamente el código fuente de startups comenzará a ser más complejo para poder paralelizar tareas y hacerlas más fácil para la IA.

Feedback loop II: Testing

Recomendaciones:

- Monta desde el principio el testing tanto unitario como end-to-end.
- Usa el MCP de playwright para que la IA pueda testear mejor:
- Le pides a la IA que haga un test sobre la web y que luego **lo guarde en un fichero de test**

Feedback loop III: Multimodales

Puedes hacer lo mismo con diseño en front si tienes modelos multimodales:

- Implementas (con IA)
- Pantallazo del componente (en distintos estados)
- Fixeas errores visuales.

`whoami`

- Psychology & Computer Science
- Founder @ mindmakers (Ed-tech: aprendizaje personalizado)
- Consultor de IA y de optimización de procesos (usando IA)

Si quieres contactarme:

alex@mindmakers.es

Otras cosas interesantes

- opencode: Claude Code pero genérico (se puede usar con otros vendors)
- Claude Code GitHub actions (trabajo asíncrono)
- OpenAI Codex (en OpenAI for teams): trabajo asíncrono en ramas.
  - Codex también es el nombre del CLI oficial de OpenAI (soporta también Google Gemini y otros)
- Daré una charla el jueves 26 en Valencia sobre IA. Habrá otro ponente hablando de ello para que podáis ver dos perspectivas diferentes.