

*PythonBrasil*²⁰₂₆

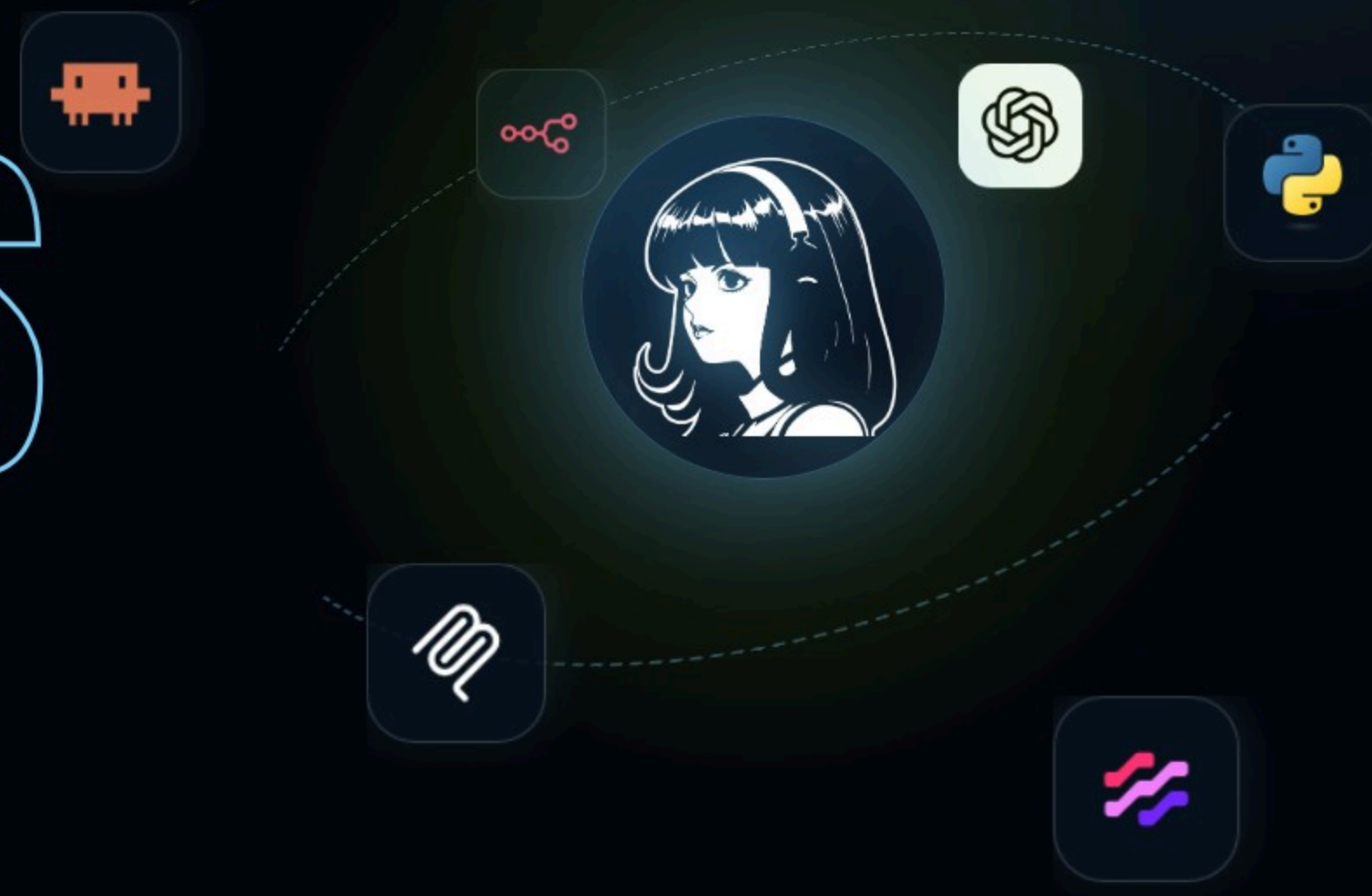
AGENTES AUTÔNOMOS

de IA com **Hermes Agent**

Linguagem natural, memória e
ferramentas conectadas à operação.

Felipe Azambuja

Hermes Agent · Nous Research e comunidade



FELIPE AZAMBUJA

Programador, professor de programação e
fundador da **PycodeBR**

• Python

• Software

• Inteligência Artificial



PALESTRANTE

Felipe Azambuja

PycodeBR

PythonBrasil 2026





PycodeBR

*Escola online de programação
e Inteligência Artificial.*

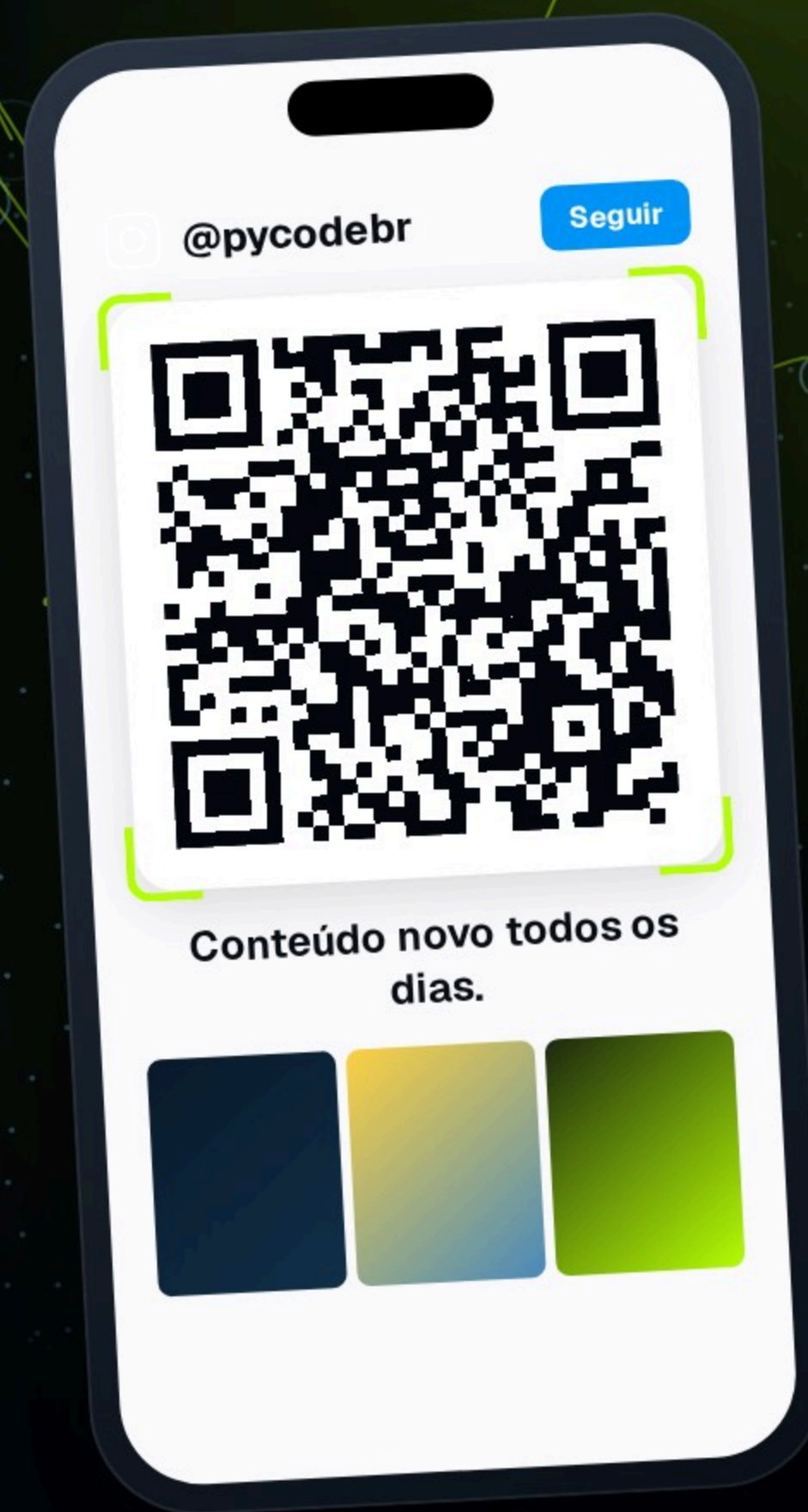
</> Programação

≡ Sistemas

⌘ IA aplicada

+8 mil

alunos ativos no Brasil
e no mundo



Acompanhe os slides no seu dispositivo.

Aponte a câmera para o código e navegue no seu ritmo.

pycodebr.com.br/python-brasil-2026/

Apresentação, downloads e materiais de estudo.



2022

2023

2024

2025

2026

2022: ChatGPT e aplicações com modelos de linguagem

**ChatGPT**

30 de novembro

Explique este código em Python.

```
for item in range(3): print(item)
```



O laço percorre 0, 1 e 2 e imprime cada valor. Quer ver uma variação com lista?

Sim, agora com uma lista de nomes.

A interface de conversa chega ao cotidiano.

**LangChain**

Primeiro pacote Python: 24 de outubro

```
$ pip install langchain
```

Aplicação

Seu código e suas regras

Modelo

Uma chamada por API

Dados

Documentos e ferramentas

O desenvolvedor conecta o modelo à aplicação.

2022

2023

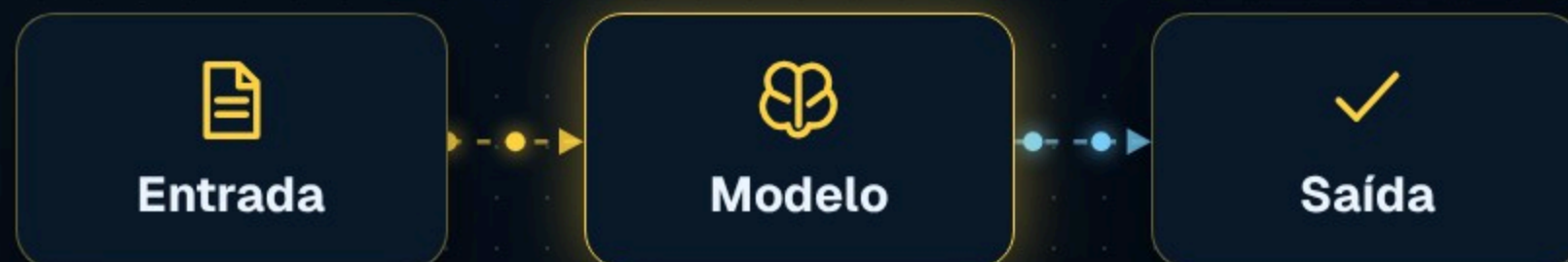
2024

2025

2026

2023: frameworks, fluxos visuais e agentes

Construir um fluxo



LangChain e Langflow ajudam a compor aplicações. O n8n integra LangChain em outubro.

Experimentar agentes



O modelo participa da escolha das ações e recebe os resultados das ferramentas.

- Workflows e agentes podem coexistir. O n8n já existia desde 2019.

2022

2023

2024

2025

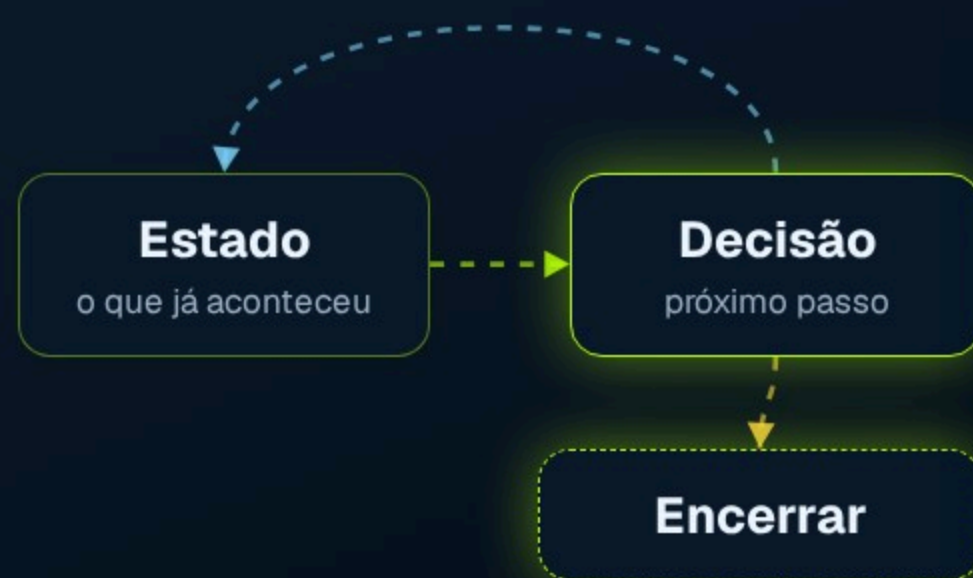
2026

2024: estado, ciclos e integrações com MCP



LangGraph

Janeiro



Continuar, repetir ou encerrar

O fluxo guarda o que já aconteceu.



Langflow 1.0

24 de junho

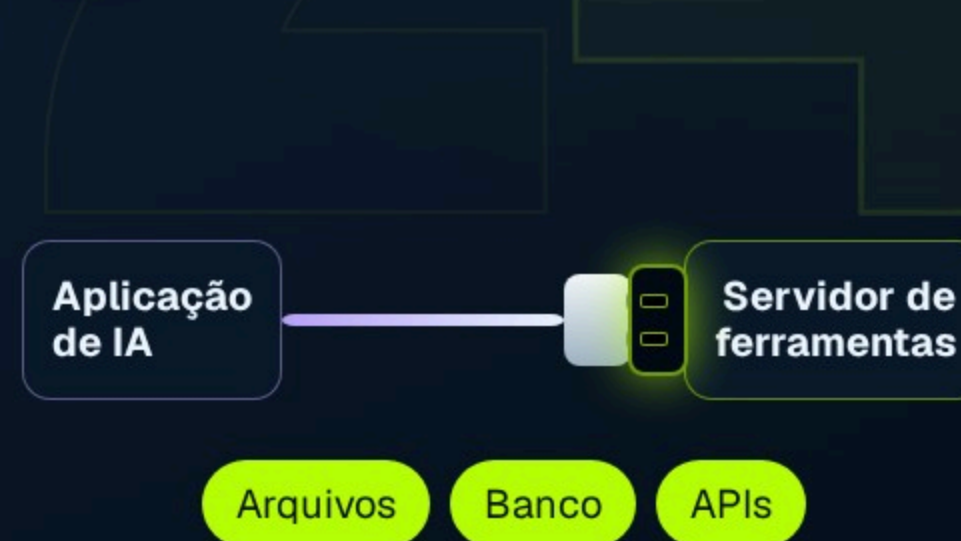


A composição da aplicação também pode ser visual.



MCP

25 de novembro



Um protocolo para conectar ferramentas e dados.

- Model Context Protocol padroniza a integração; as permissões continuam na implementação.

2022

2023

2024

2025

2026

2025: a IA passa a trabalhar no projeto

projeto · terminal

projeto/

└ src/busca.py

└

testes/test_busca.py

└ SKILL.md

> Corrija o bug e confira os testes.

> lendo src/busca.py

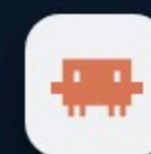
- if filtro and termo:

+ if filtro is not None and termo:

> pytest · 12 passed

> diff pronto para revisão

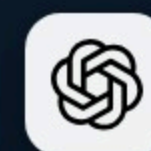
Representação didática: ler arquivos, alterar código, testar e revisar o diff.



Claude Code

Preview em fevereiro.

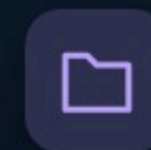
Disponibilidade geral em maio.



Codex CLI

Apresentado em abril.

Execução no terminal.



Agent Skills

Procedimentos, scripts e referências carregados conforme a tarefa.

- A CLI é o aplicativo que conduz a execução. O modelo é uma escolha separada.

2022

2023

2024

2025

2026

2026: assistentes pessoais com memória e canais

Janeiro · popularização e renomeações

~~Clawdbot~~ → ~~Moltbot~~ → **OpenClaw**



Um dos pioneiros desta onda de assistentes pessoais open source.

Seu computador

Seus canais

Ferramentas autorizadas

25 de fevereiro · Nous Research

HERMES AGENT



Execução por ferramentas, memória persistida e criação de skills.

Núcleo em Python

Continuidade

Auto skills

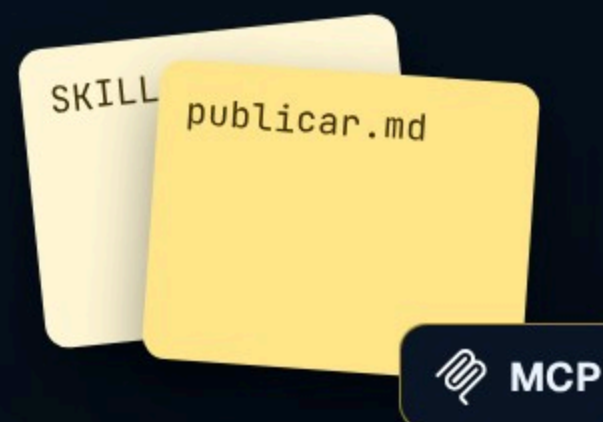
• O projeto que se tornou OpenClaw começou em 2025. Hermes é um projeto independente, lançado depois.

CLIs, skills, MCPs e agentes autônomos

```
$ claude "corrija o teste"
editando src/ · rodando testes
```

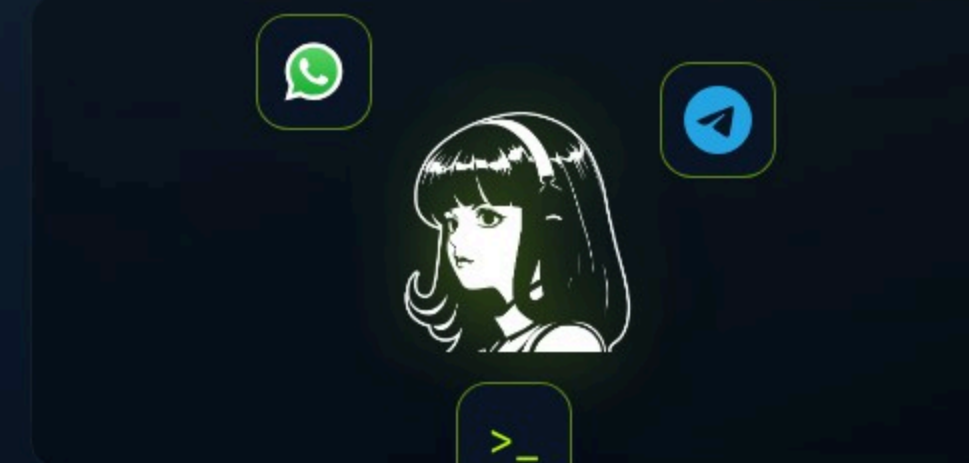
CLIs de programação

A IA trabalha dentro do projeto.



Skills e MCPs

Procedimentos reutilizáveis e integrações.



Agentes autônomos

Contexto e execução entre tarefas e canais.



Combinados no mesmo processo

Um Hermes consulta uma skill, chama uma ferramenta MCP e usa uma CLI no código.

Linguagem natural para executar tarefas com o Hermes



- O runtime executa as ferramentas e devolve os resultados; o modelo participa da escolha da próxima ação.

Memória e auto skills no Hermes

MEMORY.md

Fatos persistidos

Informações selecionadas sobre você e sua operação.

Exemplo: formato e regras das entregas.

Base de contexto

Detalhes consultáveis

Documentos, projetos e histórico recuperados quando fazem falta.

Exemplo: fontes e decisões de uma palestra.

SKILL.md

Procedimentos revisados

```
## Publicar uma página
1. Gerar os arquivos
2. Enviar ao destino autorizado
3. Conferir a versão servida antes de concluir
```

Primeira tarefa

Próxima tarefa



Hermes

✓ Publicar a atualização

✓ Recuperar a skill de publicação

✓ Comparar a versão servida com o arquivo local

✓ Entrega conferida

Na próxima publicação, o Hermes consulta essa skill e compara a versão servida com o arquivo local antes de concluir.

- O aprendizado fica em arquivos e contexto recuperáveis, sem alterar os pesos do modelo.

Infraestrutura de IA da PycodeBR

Acessos

Contexto

Integrações

Entregas



O Hermes usa modelos e consulta serviços por APIs, MCPs ou Composio. Cada conexão precisa de autenticação e permissões próprias.

- Essa base pode ser replicada e adaptada a outras operações, conforme sistemas, recursos e permissões.

Rotinas, eventos e pedidos na operação



REGISTRO DA OPERAÇÃO

23:30

23:00 Sincronização da base
commit e push no repositório

22:00 Reconciliação de vendas
divergências sinalizadas

19:12 Alerta de falhas nas rotinas
aviso no canal da operação

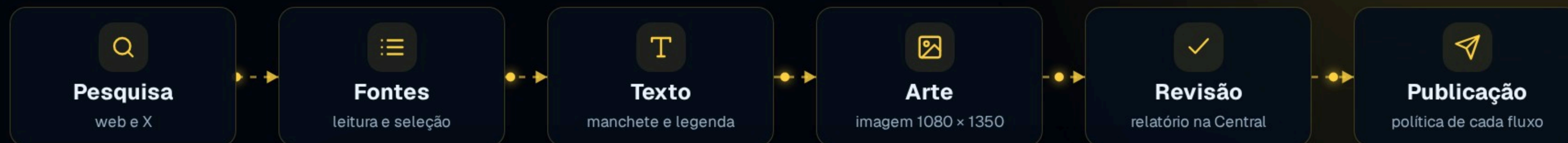
17:24 Pedido no Slack: criativos da campanha
propostas para revisão

15:00 Publicação aprovada
post no horário definido

● Agenda · horário definido ● Evento · webhook ● Pedido · conversa

● Exemplos da operação PycodeBR. Cada rotina tem permissões, saída e verificação próprias.

Pipeline de conteúdo para o Instagram



POSTS PUBLICADOS PELA PYCODEBR



Fluxo A ^{2 posts}
publica nos horários definidos

Fluxo B ^{2 posts}
aguarda a aprovação da equipe

Todos os dias, com pesquisa, texto e arte no mesmo pipeline.

- Geração, revisão e publicação têm critérios próprios.

Painéis e materiais sob demanda

PERGUNTA DA OPERAÇÃO

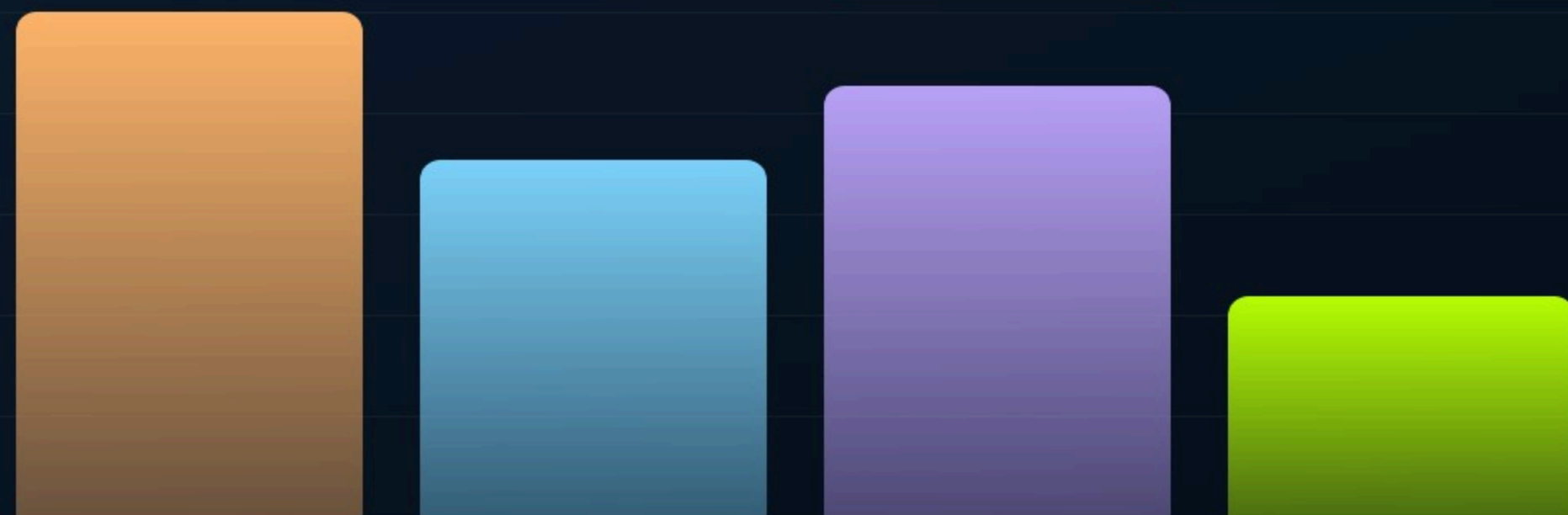
Qual canal trouxe mais inscrições nesta semana?

Semana atual

Por canal

Inscrições

fonte indisponível



Orgânico

E-mail

Anúncios

Indicação

Coleta: 3 de 4 fontes

Total parcial · não publicar como completo

Uma fonte indisponível aparece como indisponível, nunca como zero.

AGENTES
AUTÔNOMOS

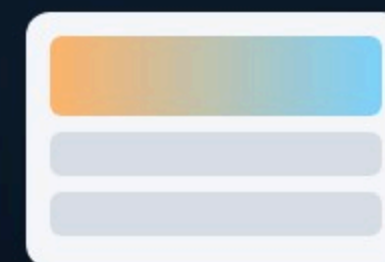
Apresentações

- links abrem
- texto legível

Guia
Passo 1
- instalar
- testar

Guias

- arquivos completos
- fontes citadas



Páginas e relatórios

- acesso adequado
- versão servida conferida

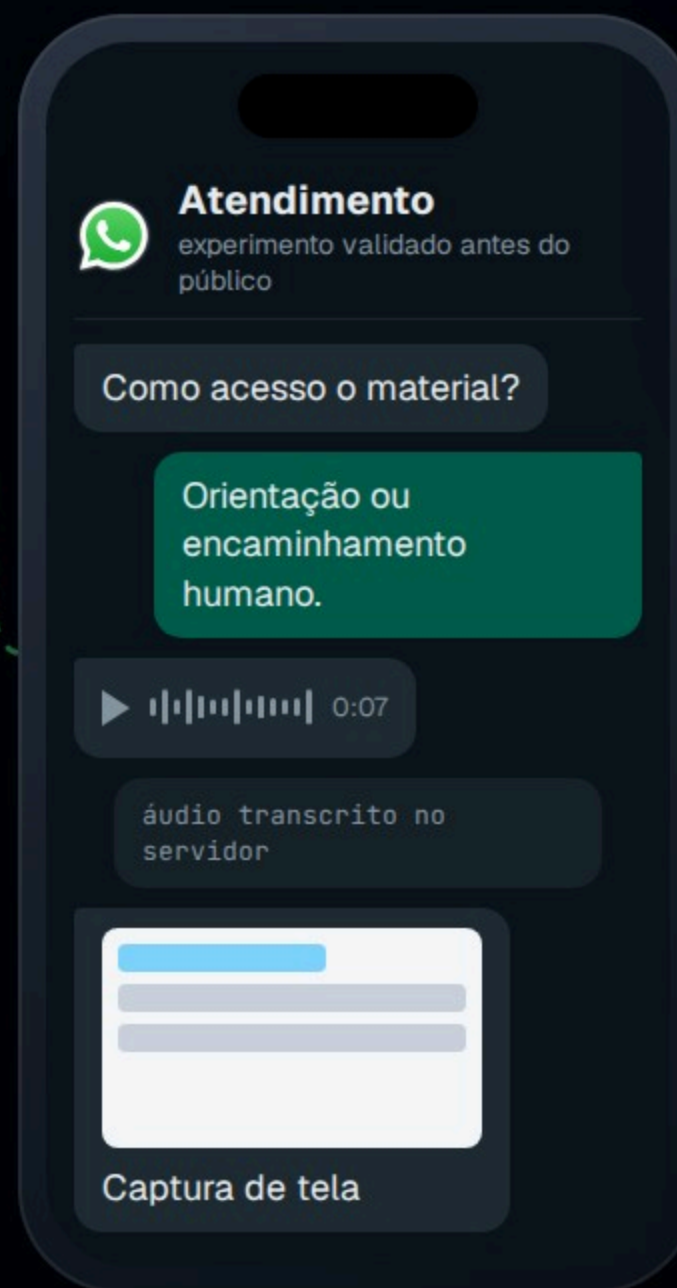
- Gráfico e cartões ilustrativos, sem métricas da empresa. Cada formato tem uma conferência própria.

Atendimento e proatividade nos canais da operação

CONTEXTO MULTIMODAL



Algumas etapas rodam com código, sem chamar o modelo a cada mensagem.



INICIATIVA AUTORIZADA

EQUIPE

A campanha precisa de criativos novos.

KRATOS

Preparei dez propostas para revisão.

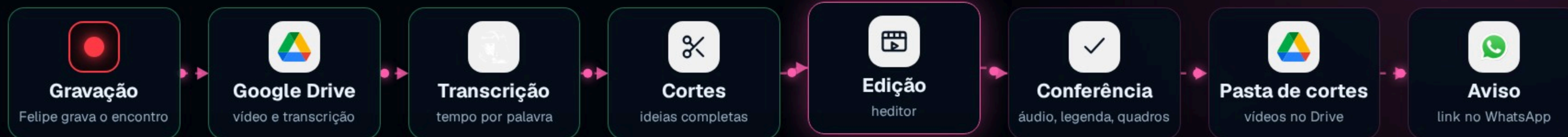


REVISÃO DA EQUIPE

Preparar a proposta era a ação permitida. Publicar e comprar mídia são decisões separadas.

● Interfaces ilustrativas. A coleta e a iniciativa dependem das permissões de cada fluxo.

Cortes de vídeo: da gravação ao aviso no WhatsApp



Cada corte ganha um mundo visual próprio, montado em HTML com GSAP e renderizado com HyperFrames.

Cinema

Game

Thriller

IDE

Aeroporto

Chat

Documentário

Podcast

Missão

`heditor` · `skill pública da PycodeBR` · `github.com/pycodebr/heditor`



● Fluxo usado nos cortes dos encontros Elite. Trechos reais editados pelo Hermes, exibidos sem áudio.

Workflow de IA Assistida em cinco passos

01
Refinar o pedido

02
Gerar o PRD.md

03
Preparar o template

04
Executar a sprint

05
Revisar, corrigir e
versionar

```
- condicoes = request.GET
+ condicoes = filtrar_campos(request.GET)
pytest · 18 passed
$ git commit -m "feat: salva filtros de busca"
$ git push

🔗 próxima sprint
```

05

Sprint → Review, Correções e Commit

Revise, teste e corrija antes do commit e do push. Depois, avance à próxima sprint.



Commit validado

O ciclo retorna à próxima sprint.

- O quinto passo se repete a cada sprint. O deploy é planejado no pedido e no PRD e acontece depois da validação.

Hermes como harness e orquestrador de desenvolvimento



- `delegate_task` cria filhos Hermes. As CLIs externas são chamadas pelo terminal, com ambientes e acessos preparados.

Deploy e observabilidade da aplicação



- Base dos encontros Elite #03 e #04. cAdvisor e node-exporter complementam as métricas; Alloy é a recomendação atual para a coleta de logs.

Arquitetura de monitoria com Hermes e MCPs



- Arquitetura de referência. O alerta inicia a análise; os MCPs fornecem consultas. Gráficos ilustrativos, sem dados de produção.

Reports de usuários e propostas de correção



• Arquitetura de referência proposta. Eventos e consultas são mecanismos separados; a cena não cria PRs nem altera serviços.

Aprovação da PR, merge e deploy da correção



Corrigir a combinação de filtros

- ✓ Teste do comportamento afetado
- ✓ Diff revisável e riscos descritos
- ✓ Versão vinculada à aprovação

```
- if filtro and termo:  
+ if filtro is not None and termo:
```

Merged



Privado de Felipe

mensagem do agente

Preparei uma PR com a correção, os testes e os riscos. Você aprova esta revisão?

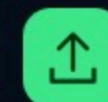
Aprovado.

Aprovar esta revisão

Pedir ajustes



Merge autorizado



Deploy conferido



Correção conferida



Simular a entrega

Recomeçar

Percurso ilustrativo concluído. A versão implantada foi conferida no exemplo.

- Demonstração local, sem enviar mensagens nem alterar repositórios. Uma mudança na PR exige nova revisão.

Permissões e limites para a execução autônoma

Dentro do escopo autorizado

Consultar métricas e logs

Investigar e testar em isolamento

Preparar uma branch e uma PR

Orçamento, limites de tempo e rastreabilidade.



APROVAÇÃO

Com aprovação definida

Merge da revisão aprovada

Deploy no ambiente combinado

Mudanças sensíveis de dados e acesso

Controles nas ferramentas, além das instruções.

- Mensagens, reports e logs são entradas externas. O conteúdo deles ajuda na investigação e continua sujeito às regras de autorização.

Continue com os materiais da palestra.

Começando com Hermes do zero novο

instalação, primeiros prompts, Composio e base de contexto

Workflow de IA Assistida

os cinco passos e o ciclo por sprint

Arquitetura e observabilidade

operação, monitoria e manutenção com revisão humana

Laboratório em Python

simulação local de aprovação, merge e deploy

 github.com/pycodebr/pycodebr_python_brasil_2026

```
$ git clone https://github.com/pycodebr/pycodebr_python_brasil_2026
guia: materiais/comecando-com-hermes-do-zero.md
```



Repositório com slides, guias e código

OBRIGADO · PERGUNTAS

Hermes Agent · Nous Research e comunidade

Felipe Azambuja · PycodeBR

MATERIAIS E PERGUNTAS

26 / 26