

PRODUCT DESIGN COM IA

Do protótipo ao produto.

Como a IA mudou minha metodologia de entrega e me colocou no estratégico, não só no visual e nas telas

A PERGUNTA

E se o designer não precisasse mais entregar um protótipo para alguém reconstruir tudo em código?

ANTES



Créditos



Figma



Lovable



Figma Make



Protótipo



Handoff



O PROJETO SAIU DA FERRAMENTA

AGORA



GitHub



Código



Ambiente local



Codex



Claude Code



Devin



Produto funcional

Os produtos que eu construo

Product Designer na theGarage IA, atuando na criação e na evolução de produtos digitais em estágios diferentes.

Projeto grande exige contexto acumulado e nenhuma ferramenta segurava esse contexto por muito tempo.

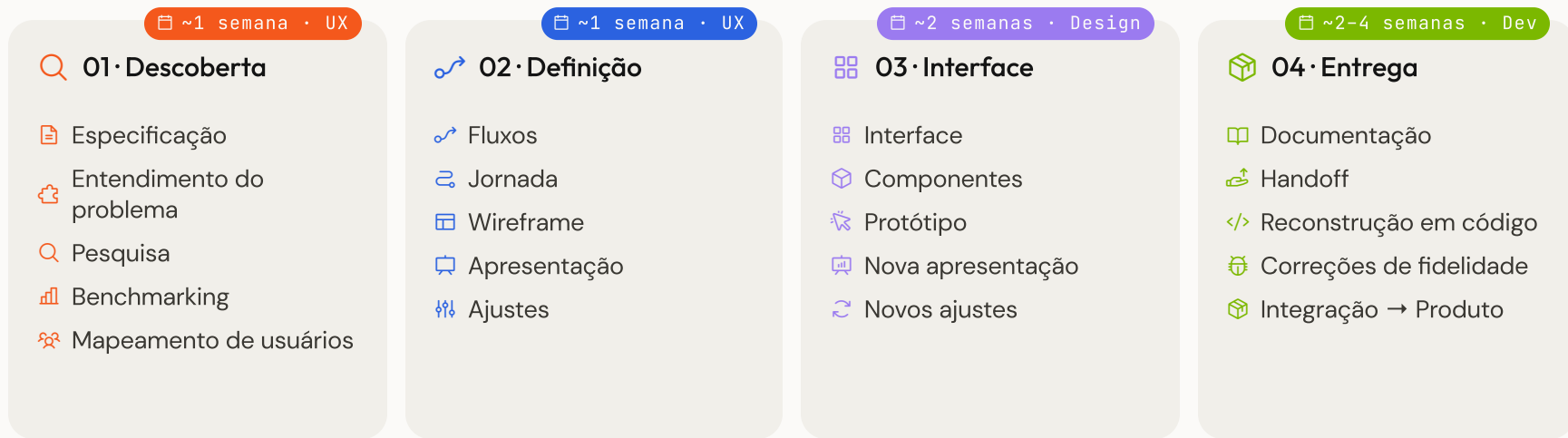
Cada novo fluxo multiplicava telas, estados e horas de retrabalho.

E quanto mais o produto evoluía, mais lento era estar preso a uma única plataforma.



Meu portfólio: desenhado e publicado por mim, direto em código — cases, experiências e contato, sem etapa de reconstrução.

Pesquisar, rascunhar, desenhar, validar, prototipar e só então reconstruir tudo em código



O TEMPO PERDIDO EM CADA PASSAGEM DE CONTEXTO

Perda de contexto

Interpretação errada

Retrabalho

Decisão duplicada

Mudança em dois lugares

Atraso

4 a 8 semanas até o produto. Longo não por ser completo — longo por reconstruir o mesmo produto várias vezes.

Depois de todo o trabalho, o produto ainda não existia



Para o stakeholder, o protótipo parecia pronto.



Para o desenvolvimento, começávamos tudo de novo, ainda sem as interações reais.



Interações limitadas



Dados não reais



Design System incompletos



Responsividade aproximada



Regras condicionais limitadas



Fluxos extensos difíceis de manter



Sem integrações reais



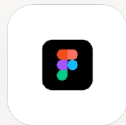
Comportamento explicado em doc



Reconstrução completa em código



...e depois, correção de fidelidade



O Figma não sai do meu dia a dia. Ele só deixou de ser onde o produto nasce.

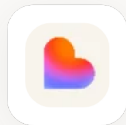
INSUBSTITUÍVEL NO DIA A DIA DO TIME DE PRODUTO

- ✓ Exploração visual
- ✓ Colaboração e workshops (FigJam)
- ✓ Design system e biblioteca
- ✓ Alinhamento rápido com o time
- ✓ Rabiscos rápidos para validar ideia

MAS O PROTÓTIPO, COMO ENTREGA FINAL, AINDA

- ⊗ Precisa ser interpretado
- ⊗ Precisa ser reconstruído em código
- ⊗ Depende de handoff e documentação
- ⊗ Fica longe de dados e integrações reais
- ⊗ Não expressa comportamento de produção

Não é sobre cancelar o Figma, ele segue poderoso, hoje ainda mais com MCP e agentes. É sobre não fazer o produto viver e morrer dentro de um protótipo.



A ferramenta é inteligente. O processo segue preso.

Não adianta trocar o Figma pelo Lovable.
Trocamos de ecossistema, mas continuamos presos a um.



Créditos próprios da
plataforma



Modelo escolhido pela
ferramenta



Limites de contexto



Restrição para alterações
grandes



Novo ciclo, nova espera



Menor controle de arquivos e
histórico



Dificuldade de alternar entre
IAs



Evolução condicionada às
regras do ambiente



Trocar de plataforma muda o
formato da dependência

A dor do dia a dia

❓ Perguntas antes de cada prompt

- ❓ Quantos créditos ainda tenho?
- ❓ Este prompt vale o limite?
- ❓ Posso testar algo aleatório?
- ❓ Devo explorar ou aceitar a primeira solução?
- ❓ É melhor guardar crédito para outro projeto?

💰 Créditos do mês



Cada barra é uma tentativa. O comportamento que isso cria não é exploração é contenção.

Eu tinha uma ferramenta criada para acelerar a experimentação, mas evitava experimentar para não gastar créditos.

Já assinávamos as melhores IAs do mercado. E o projeto continuava preso a uma delas.

EFEITO NO DIA A DIA

Trabalho pausado

Cada limite atingido virava espera por liberação

 ChatGPT + Codex

 Claude + Claude Code

 Devin

 Lovable

 Figma + Figma Make

CONTRATADO

Ferramentas já pagas

Várias assinaturas de IA disponíveis para o time.



INTERRUPÇÃO

Fila de liberação

Mesmo com as assinaturas, o trabalho parava até liberar o próximo ciclo.



SUBUTILIZADO

Capacidade parada

As outras IAs não podiam atuar livremente sobre o mesmo projeto.



O problema não era a falta de IA. Era o projeto estar no lugar errado.

Por que eu media o quanto eu podia pensar?

Eu não estava insatisfeito com a ferramenta, e sim com um dia a dia em que eu media o quanto podia pensar. Até perceber que o problema não era qual ferramenta usar — era onde o projeto estava guardado.



A limitação não era técnica.
Era criativa: eu parava de explorar antes de esgotar as alternativas.



Quanto mais contexto eu queria dar sobre o produto, mais o processo travava — ele punia justamente contexto e iteração.



A pergunta mudou: de “qual ferramenta usar?” para “onde o projeto precisa morar para eu ter liberdade?”. A resposta: no repositório da empresa.

O projeto é da empresa. Não é mais refém de nenhum ecossistema.

Um repositório oficial virou a camada de liberdade entre o produto e qualquer ferramenta de IA.

01 • ORIGEM

O projeto pode começar direto em código, sem pular pesquisa e benchmarking, só de forma mais assertiva.

02 • CONEXÃO

É conectado ou transferido para o repositório da empresa.

CENTRO DO PROCESSO



GitHub + Azure DevOps

Repositório clonado na máquina, produto rodando local, histórico versionado e reversível.

clone

npm

commit

push

03 • TRABALHO

Qualquer IA atua sobre a mesma base.



Codex



Claude Code



Devin

04 • RETORNO

Push no GitHub e os ambientes conectados recebem conforme a configuração.

As ferramentas são intercambiáveis. **O repositório permanece.**

De dependente da ferramenta a **dono do processo**

⌚ ANTES

- ⌚ Projeto preso à plataforma
- ⌚ Linguagem definido pela ferramenta
- ⌚ Novos ciclos negociados a cada mês
- ⌚ Trabalho interrompido no limite
- ⌚ Menos liberdade para explorar
- ⌚ Outras assinaturas subutilizadas
- ⌚ Alterações condicionadas ao limite
- ⌚ Entrega final: protótipo

⚡ AGORA

- ✓ Projeto centralizado no GitHub ou Azure Devops
- ✓ Execução local
- ✓ IA escolhida conforme a tarefa
- ✓ **Qualquer IA, qualquer modelo**
- ✓ Teste antes de publicar
- ✓ Histórico preservado e reversível
- ✓ Continuidade quando uma IA atinge o limite
- ✓ **Nenhuma parada por liberação em 3 meses**
- ✓ Entrega funcional, navegável e responsiva

Antes eu escolhia o que fazer com base nos créditos. Agora escolho a IA com base no problema.

Crédito esgotado deixou de parar o trabalho

Posso começar no Codex, sigo no Claude Code, termino no Devin, mesma base, sem recomeçar. A escolha é técnica, não de saldo.



Codex

Pesquisa e planejamento
Crítica de UX
Implementação e bugs
Componentes e revisão



Claude Code

Contextos extensos
Muitos arquivos de uma vez
Refatoração e estrutura
Coerência entre partes



Devin + Windsurf

Navegação no repositório
Investigação de problemas
Implementações longas
Execução de tarefas



Lovable

Início de projeto
Ambiente visual
Publicação
Sem controle exclusivo da evolução

Liberdade total sobre o projeto: qualquer IA, qualquer modelo.

A empresa não comprou uma nova solução, conectou melhor as que já pagava.

Não pulamos etapas. Reduzimos a distância entre elas.

 ~1 dia

01 • Descobrir

Quem são os competidores
Referências visuais do mercado
Direção visual definida
Design system já começa aqui

 ~1 dia

02 • Definição

Especificação do produto em mãos
Necessidade, dor e objetivo mapeados
Sem pesquisa longa com usuário
Hipótese validada em minutos

 ~3-5 dias

03 • Desenhar

Interfaces construídas na base real
Componentes reutilizáveis
Fluxos e casos extremos
Validação com time e cliente

 ~2 dias

04 • Validar

Dados reais, nada de dados fake
Usabilidade e fluxos testados
Riscos identificados antes
Analytics e métricas quando quiser

COMPARADO AO FLUXO TRADICIONAL



Mais rápido



Mais fiel



Cenários reais



Sem reconstrução

De 4 a 8 semanas para 1 a 2 semanas.
Com mais qualidade: testamos cenários reais, não protótipos mocados.

De especificador a construtor da experiência

DESIGNER RAIZ TAMBÉM OLHA CÓDIGO

ANTES

-  Wireframes e telas
-  Interações simuladas
-  Protótipos
-  Documentação de comportamento
-  Handoff
-  Acompanhamento da reconstrução
-  Muitos ajustes e validações — muitas horas

Muito tempo investido para chegar em um protótipo estático — que pode nunca gerar valor.

AGORA



Pesquisa e direção



Design system estruturado



Componentes reais



Fluxos implementados



Responsividade validada



Estados testados



Trabalho no repositório



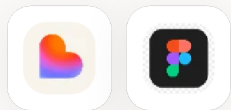
Orquestração de IAs



Tempo e eficiência



Usabilidade no produto real



Hoje eu não preciso morar no ecossistema de ninguém

Dentro do Lovable ou do Figma, o projeto ficava preso às regras do ambiente. Agora ele é meu — e evolui com qualquer IA, qualquer modelo, sem trocar de casa.

Liberdade não é escolher a melhor ferramenta. É não depender de nenhuma para o projeto seguir.



O repositório já nasce no GitHub



O design system é implementado desde o início



A navegação já é real



Os componentes já são reutilizáveis



Responsividade e estados já podem ser testados



Não existe reconstrução posterior do protótipo



Autonomia



Liberdade



Criatividade



Tempo



Dinheiro

O designer não substitui o desenvolvedor. Ele reduz o que precisa ser reinterpretado.

O designer entrega

-  Interface funcional
-  Componentes
-  Navegação e estados
-  Formulários
-  Responsividade
-  Dados simulados
-  Design system implementado
-  Base versionada

Engenharia segue essencial

- Backend
- Banco de dados
- APIs
- Arquitetura
- Segurança
- Autenticação
- Permissões
- Performance
- Testes
- Observabilidade
- Escalabilidade
- Qualidade técnica

Antes recebia

-  Arquivo visual e protótipo limitado
-  Documentação para interpretar
-  Interface para reconstruir

Agora recebe

-  Interface funcional e componentes
-  Estados e navegação mapeados
-  Repositório versionado para revisar

O que mudou em tempo, entrega e autonomia

TRÊS MESES

1-2 sem**Do briefing ao funcional**

Antes: 4 a 8 semanas.

5x**Velocidade de ajuste**

Revisto no mesmo dia.

zero**Interrupções por limite**

Nenhuma pausa por liberação.

3 meses**Autonomia operacional**

Sem aprovação externa.

 Operacional

Menos interrupções

Menos solicitações internas e espera

Mais continuidade no trabalho

 Retrabalho

Menos reconstrução entre protótipo e código

Menos documentação só para handoff

Menos correção de fidelidade

 Autonomia

Liberdade para testar

Escolha entre diferentes IAs

Controle do repositório e continuidade

O ganho visível foi de tempo. O mais relevante foi de interrupção, retrabalho e dependência.

Começou como ganho de tempo. Virou autonomia.

A mudança não foi trocar uma ferramenta por outra. Foi garantir que nenhuma delas fosse dona do processo.

GitHub como centro

IA por contexto, não por saldo

UX preservado

Entrega funcional