

Por que a IA repete o componente genérico

(e o que muda quando você especifica)

TDC Community · TDC Talks · quarta, 21/10/2026 · 19h às 20h30 (Brasília) · Thiago Xikota

Hoje você vai apostar, testar e sair com uma decisão pronta para um PR. Precisa só de navegador. Editor de código e ferramenta de IA são opcionais. Nenhum convite real é enviado.

Antes de começar

- 1. Abra **thiagoxikota.github.io/tdc-convite** no navegador do computador. A prática precisa de teclado; no celular fica difícil. Sem internet, extraia o kit-participante.zip e abra index.html.
- 2. **Não abra a saída A ainda.** A primeira atividade é uma aposta.
- 3. Se algum arquivo não abrir, acompanhe pela tela do facilitador. Todas as atividades funcionam assim.

Aposta 1: bingo do componente genérico

O pedido A, na parte de produto, era só isto: *"Crie um componente moderno e responsivo para convidar uma pessoa para uma equipe."* Marque o que você acha que veio. Três das doze casas são armadilhas: não vieram.

☐ Sobretítulo em caixa alta	☐ Campo de mensagem pessoal	☐ Título motivacional	☐ Frase de efeito no card
☐ Card no centro, com sombra	☐ Foto ou avatar da pessoa	☐ Ícone num quadradinho	☐ Botão de largura total
☐ Setinha no botão	☐ Link de convite para copiar	☐ Tela de sucesso com check	☐ Membro / Administrador

Casas marcadas: ____ de 12.

Aposta 2: quantos testes A passa?

Chute antes: 0, 1, 2 ou 3. Meu chute: ____

Abra demonstracao/convite-a.html. O **Painel do laboratório**, embaixo do formulário, mostra quantas chamadas a API recebeu, o último e-mail, o perfil e o estado. Recarregue a página antes de cada teste.

Teste	Faça	Passa?
C10 contexto	Olhe a página. Ela cabe dentro de uma gestão de pessoas que já existe?	

Teste	Faça	Passa?
C3 espera	Envie novo@example.com e, durante 1,2 s, aperte Enter e clique de novo. Pannel: quantas chamadas?	
C6 falha	Envie falha@example.com. Sem recarregar, envie de novo.	

Anote uma decisão que A tomou sem ninguém pedir: _____

A saída B e a decisão D1

O pedido B (demonstracao/pedido-b.txt) traz contexto, perfis Leitor e Editor, dez critérios e uma decisão de produto, D1: *depois do sucesso, editar e-mail ou perfil limpa a confirmação anterior*. Abra demonstracao/convite-b.html.

Teste D1: envie ana@example.com, espere a confirmação e comece a apagar o e-mail. A frase sumiu? _____

Aposta 3: a janela

Envie ana@example.com com Leitor e, **durante a espera**, troque o perfil para Editor. Para dar tempo: **Enter, Tab e a tecla E**. O que aparece?

A. A confirmação nem aparece · B. Aparece "perfil Editor" · C. Aparece "perfil Leitor" com o select em Editor · D. Dá erro

Minha letra: ____ O que aconteceu: _____

Faça o mesmo em A. O que muda? _____

Prática: dê um dono à janela (14 min)

Duas regras legítimas. Você pode propor uma terceira, desde que diga o custo.

Regra	O que faz	Custo
Travar	Campos travados enquanto o convite está a caminho.	Quem errou uma letra espera a resposta. Em 1,2 s, nada; numa espera de 40 s, irrita.
Recibo	Deixa editar. A resposta diz de quem ela é e avisa que os campos mudaram.	Mais estado no componente e uma frase mais longa para ler.

Seu cartão da janela, já como PR (5 min)

Título: Convite: a resposta fala do envio dela
Janela: a pessoa muda e-mail ou perfil com o convite a caminho
Regra: [o que a tela faz]
Custo: [o que perdemos com essa regra]
Teste: envia ana@example.com como Leitor, Enter, Tab, E
→ [resultado esperado]
Hoje: falha em convite-b.html · passa no patch da minha regra

A linha "Hoje" é a mais forte do PR: um teste que falha agora, com passo a passo. No seu repo, troque arquivos e funções pelos reais.

Rode o seu teste (8 min)

Arquivo	O que esperar	Resultado
demonstracao/convite-b.html	Falha: a tela fica ambígua	
kit-tdc/b-patch-trava.html	Passa se você escolheu travar	
kit-tdc/b-patch-recibo.html	Passa se você escolheu recibo	

Escolheu outra regra? Confirme que B falha e escreva o resultado esperado. Esse teste vira o seu PR.

Aposta 4: a busca

```
busca.addEventListener('input', async () => {
  const termo = busca.value;
  const pessoas = await buscarPessoas(termo);
  lista.render(pessoas, termo);
});
```

A equipe tem Ana, Anderson e Luana. Você digita "an" e depois "ana". A rede responde "an" em 900 ms e "ana" em 200 ms. O que fica na tela no fim?

Qual regra você usaria aqui? ☐ trava ☐ recibo ☐ descarta a resposta velha ☐ outra: _____

Por quê? _____

Três perguntas para o próximo PR

1. Quais janelas esse componente tem (antes, durante, depois), e quem decidiu cada uma?
2. Essa resposta é de leitura ou de escrita?
3. Que teste falha se alguém mudar a regra?

Para casa

Com IA: cole seu cartão no fim de demonstracao/pedido-b.txt, use conversa nova e espere no máximo 4 minutos. Salve a primeira saída como meu-convite.html, registre a ferramenta e rode o mesmo teste. Se a IA alterar a API simulada ou o painel, o resultado não é comparável.

Respostas das apostas, o código e os patches explicados: thiagoxikota.github.io/tdc-convite/respostas.html, depois da aula.

Procedência e limites

De onde vêm A e B. Duas sessões separadas do Codex em 19/10/2026, uma geração por condição, mesma base, primeira saída guardada sem edição. O modelo não foi informado pela ferramenta. Pedidos integrais e registro em demonstracao/. Um par de saídas não mede com que frequência isso acontece; a evidência de repetição usada na aula é a previsão da própria sala.

O que os patches e a referência são. kit-tdc/b-patch-trava.html e kit-tdc/b-patch-recibo.html são cópias de B com patches escritos à mão para a aula, rotulados na própria página. kit-tdc/referencia.html é uma implementação didática completa. Nenhum deles é saída gerada.

O que o laboratório não prova. A API simulada (window.apiConvite, 1,2 s de espera) não tem rede, servidor nem autorização: aceita o perfil que recebe. Bloquear o botão não protege contra recarga ou resposta perdida; isso pede um contrato com o backend, como uma chave de idempotência. falha@example.com simula só uma falha conhecida. Leitor de tela e zoom nativo não foram testados.

Os dez critérios do pedido B, para consulta

ID	Critério
C1	E-mail vazio ou inválido não chama a API; erro junto ao campo e foco nele.
C2	Leitor por padrão; só Leitor e Editor.
C3	Andamento visível na espera e nenhuma chamada duplicada.
C4	Confirmação mostra e-mail e perfil enviados.
C5	existe@example.com explica que a pessoa já está na equipe e preserva os campos.
C6	Falha preserva campos; nova tentativa funciona.
C7	Teclado opera tudo, com foco visível; resposta não rouba o foco de quem saiu.
C8	Labels, erro associado, aria-invalid, região de status.
C9	Sem corte em 320 px e zoom de 200%.
C10	Bloco integrado à gestão de pessoas, tokens existentes, sem hero, sidebar ou avatar.

Fontes: WCAG 2.2, teclado · W3C WAI, notificações de formulário · RFC 9110, métodos idempotentes · MDN, AbortController