

Eventos personalizados no GTM sem alterar o código do site

Kit de autonomia para mídia: triggers nativos,
Matches CSS Selector, DOM scraping — e quando parar.

Analistas de mídia e gestores de tráfego que precisam de eventos **amanhã**, enquanto engenharia está no backlog — sem transformar o funil da empresa em dívida técnica de seletores CSS.

Fábio Oliveira · Analytics & Atribuição

<https://dadosquebatem.com.br>

Guia prático · 2026

Sumário

1	Por que DIY no GTM — e o preço disso	3
1.0.1	S	3
2	Triggers nativos (sem código do site)	4
2.1	Clique: Just Links vs All Elements	4
2.2	Element Visibility	4
2.3	Formulário	4
3	Matches CSS Selector	6
4	DOM scraping	7
4.1	DOM Element Variable	7
4.2	Custom JavaScript (mínimo)	7
5	Limites que o DIY não resolve bem	9
5.1	SPA (React / Vue / Angular)	9
5.2	Shadow DOM	9
6	Do trigger ao evento no GA4	10
7	Checklist de QA (imprimível)	11
8	Quando parar e pedir Data Layer	12
8.0.1	P	12

Por que DIY no GTM — e o preço disso

A campanha sobe amanhã. Você precisa de clique no CTA, impressão do popup de saída e download do PDF. Engenharia está no backlog. O ticket de `dataLayer.push` não sai esta semana.

O Google Tag Manager existe para esse gap: triggers nativos, seletores CSS e variáveis que leem o DOM **sem** alterar o repositório do site. Funciona. Também é frágil. Uma troca de classe no redesign e o evento some em silêncio — o Ads continua otimizando como se nada tivesse acontecido.

Use GTM sem código para **destravar campanha**. Não construa o funil inteiro da empresa em cima de `div.css-xyz`. O padrão de longo prazo continua sendo o contrato de Data Layer com engenharia.

1.0.1 S

sintomas típicos

Sintoma	Causa típica
Clique no botão “não conta” Form envia, GTM calado	Clique caiu no ícone/ <code>span</code> ; trigger só olhava o texto AJAX / <code>preventDefault</code> — Form Submission nativo morto
Popup invisível para analytics Evento no lugar errado	Sem thank-you; elemento injetado depois do load <code>Click Classes + contains</code> (falso positivo)
Evento sumiu após redesign	Schema drift: seletor ou ID mudou

Autonomia sem critério vira dívida: velocidade agora, confiabilidade zero no próximo deploy de front.

Triggers nativos (sem código do site)

2.1 Clique: Just Links vs All Elements

- **Just Links** — escuta cliques em `<a>`. Use **Wait for Tags** quando a navegação é rápida demais e a tag não chega a disparar.
- **All Elements** — qualquer nó (botão, imagem, SVG, `div`). Mais amplo; combine com condição forte de seletor.

Ative as variáveis built-in: Click Text, Click Classes, Click ID, Click URL e, principalmente, **Click Element**.

2.2 Element Visibility

Para banners, popups, mensagem de “obrigado” sem redirect, cards em carrossel:

1. Trigger **Element Visibility** — ID ou CSS Selector.
2. Ajuste % **mínimo** visível (50–100%) e **tempo mínimo** on-screen (ms).
3. Frequência: uma vez por página, uma vez por elemento, ou sempre que aparecer.
4. Marque **Observe DOM changes** se o elemento é injetado depois do load (modal, erro de form, toast).

O GTM preenche percent visible e duração — úteis como parâmetros no GA4.

2.3 Formulário

Form Submission nativo só ajuda quando o `submit` sobe até o `window`. Em form AJAX, a ponte sem eng costuma ser:

- clique no botão de enviar, **mais**

-
- Element Visibility na mensagem de sucesso.

Se o form vive em iFrame ou a resposta é só de rede, DIY puro não basta — veja o artigo <https://dadosquebatem.com.br/formularios-iframes-gtm>.

Matches CSS Selector

`Click Classes` devolve uma **string** (`"btn primary cta"`). Condicionar com `contains "btn"` gera falso positivo (`nav-button-container`) e quebra se a ordem das classes mudar.

O padrão certo: variável **Click Element** + condição **Matches CSS Selector**. Isso chama `element.matches()` no nó real do DOM.

Seletor	O que faz	Uso
<code>button.cta-primary</code>	Tag <code>button</code> e classe <code>cta-primary</code>	CTA principal
<code>#nav-menu a, #nav-menu a *</code>	Link do menu ou qualquer filho	Clique no ícone conta
<code>a[href\$=".pdf"]</code>	<code>href</code> termina em <code>.pdf</code>	Downloads
<code>div#cart > a</code>	Filho direto de <code>#cart</code>	Mini-carrinho
<code>[data-product-id]</code>	Tem atributo <code>data-product-id</code>	Card de produto
<code>input[type="checkbox"]:checked</code>	Checkbox marcado no evento	Opt-in / termos

Prefira atributos `data-*` estáveis a classes geradas por CSS-in-JS (`css-1a2b3c`). Se o front só expõe classe ofuscada, você já está no limite do DIY.

4

DOM scraping

Só registrar o clique às vezes não basta — você quer `item_name` ou preço ao lado do botão.

4.1 DOM Element Variable

Método: ID ou CSS Selector. Sem nome de atributo → texto (`innerText`). Com atributo → `src`, `href`, `data-value`, etc.

4.2 Custom JavaScript (mínimo)

A variável deve ser uma **função anônima** com `return`:

```
function() {  
  var el = document.querySelector('h1.product-title');  
  if (!el) return undefined;  
  return el.innerText.trim();  
}
```

Para pegar o preço do card clicado (não o primeiro da página), suba a partir do Click Element (no GTM: {{Click Element}}):

```
function() {  
  var clickEl = {{Click Element}};  
  if (!clickEl || !clickEl.closest) return undefined;  
  var card = clickEl.closest('[data-product-id]');  
  if (!card) return undefined;  
  var priceEl = card.querySelector('.price, [data-price]');  
  if (!priceEl) return undefined;  
  var raw = (priceEl.getAttribute('data-price')  
    || priceEl.innerText || '')  
    .replace(/^[^d.,]/g, '')  
    .replace(',', ' ');  
  var n = parseFloat(raw);  
  return isNaN(n) ? undefined : n;  
}
```

Isso é tática de campanha, não arquitetura. Trocar `.price` por `.product-cost` zera o parâmetro sem alarme. Não trate scrape como fonte da verdade no `dataLayer` — use só como parâmetro da tag GA4 enquanto o contrato com eng não existe.

Leia também: <https://dadosquebatem.com.br/data-layer-devs>.

Limites que o DIY não resolve bem

5.1 SPA (React / Vue / Angular)

Um único load → um pageview clássico. O trigger **History Change** simula pageviews virtuais. No GA4, desative a parte de histórico do Enhanced Measurement se você for medir SPA pelo GTM — senão duplica.

Timing: History Change pode disparar **antes** do DOM novo; título e scrape saem stale. Guia completo de SPA fica para um post dedicado; aqui o ponto é: autonomia tem teto.

5.2 Shadow DOM

Cliques dentro de Web Components muitas vezes não aparecem em Click Text / Classes / Selector. Se o CTA “não existe” no Preview, pode ser shadow. Aí o caminho é eng expor evento ou dataLayer — não mais um seletor milagroso.

6

Do trigger ao evento no GA4

1. Trigger (Click / Visibility / Custom Event interno).
2. Tag **GA4 Event** — nome em `snake_case` estável (`cta_click`, `file_download`, `exit_intent_view`). Evite `ctaClick` / nomes de campanha da semana.
3. Parâmetros: `link_url`, `file_name`, `element_text`, `form_id` — o que for estável o bastante para reportar.
4. Preview (Tag Assistant) → DebugView → só então publicar o container.

Enhanced Measurement já cobre scroll, outbound e file download em muitos sites. Antes de criar tag paralela, confira se você não está **duplicando** o que o GA4 já coleta.

Checklist de QA (imprimível)

#	Verificação	Feito quando...
1	Condição	Click Element + Matches CSS Selector (não Classes contains)
2	Filho do link	Seletor inclui a * quando o clique cai no ícone
3	Visibility	Observe DOM changes ligado se o elemento é dinâmico
4	Form AJAX	Clique + Visibility de sucesso — ou fluxo de formulários/iFrames
5	Parâmetros	Scrape sanitizado; Number onde for valor; sem PII
6	Nome GA4	Evento <code>snake_case</code> reutilizável entre campanhas
7	Preview	Tag no elemento certo; zero falso positivo na nav
8	Pós-deploy front	Revalidar seletores após release de UI

Quando parar e pedir Data Layer

Use GTM sem código para destravar campanha. Não construa o funil inteiro em cima de classes ofuscadas.

Quando o evento **paga mídia** ou alimenta Smart Bidding, o próximo passo é um **mapa de eventos App/Web**: o que o front publica no dataLayer, ownership e QA — para o redesign não apagar sua conversão.

8.0.1 P

próximo nível no blog

- Data Layer para Devs — <https://dadosquebatem.com.br/data-layer-devs>
- Formulários e iFrames — <https://dadosquebatem.com.br/formularios-iframes-gtm>
- Erros de e-commerce no GA4 — <https://dadosquebatem.com.br/erros-ecommerce-ga4>
- Artigo-base deste guia — <https://dadosquebatem.com.br/eventos-gtm-sem-codigo>

Partimos do que você já tem no container (CTAs, forms, downloads) e separamos o que pode continuar DIY do que precisa de contrato com engenharia.

WhatsApp: <https://wa.me/5562994123173?text=Ola,+quero+o+guia+GTM+sem+codigo+e+um+diagnostico+de+eventos>

Site: <https://dadosquebatem.com.br/guia-gtm-sem-codigo>