

## Boas Práticas

O banner HTML5 deve ter todas suas referências externas como relativas. É importante que o funcionamento do banner seja compatível com o SafeFrame do **DFP (DoubleClick for Publishers)**.

A construção da peça não pode depender de qualquer outro elemento da página ou bibliotecas javascript externas. Fontes, CSS, JS, HTML e imagens devem ser enviadas no pacote da peça.

## Animações

Evite ao máximo o uso de javascripts para animações / transições, dando sempre preferência ao recurso de transições do CSS.

## Javascripts

Evite recursos que elevem o processamento da máquina do usuário, como loopings, Intervals, efeitos, etc. Isso prejudica a experiência do usuário e a navegação nas páginas.

jQuery não é recomendado para anúncios HTML5 pelas seguintes razões: Desempenho reduzido em dispositivos móveis por causa da dependência de animações baseadas em tempo. Ponderação desnecessária de arquivos: a maioria das funcionalidades da biblioteca não é necessária. Como alternativa, procure bibliotecas mais leves, como a zepto.js

Não é permitido ter referências a arquivos js e bibliotecas externas, apenas aos assets do pacote.

## Peso máximo dos anúncios

Desktop / Tablet: 150KB

## Polite Download (somente via tag de terceiros)

Todos os arquivos subsequentes ao carregamento de peças in-page, que ultrapassam os tamanhos citados acima só poderão ser descarregados após o carregamento completo da página seguindo os limites abaixo:

Desktop 2MB

## Assets

Tipos de assets suportados:

- Imagens

Aceitos: .jpg .png .gif

- CSS (.css)

- Javascript (.js)

## Bibliotecas

Os limites de peso devem ser respeitados, portanto recomendamos que, se for extremamente necessário, usem bibliotecas simplificadas, que tenham apenas os recursos necessários para o funcionamento da peça.

## CLICKTAG

Existem diversas maneiras de criar a clickTag (área de clique do anúncio) e em todas elas a página do anunciante deve abrir em uma nova janela ou aba do navegador.

*Ex. HTML + Javascript:*

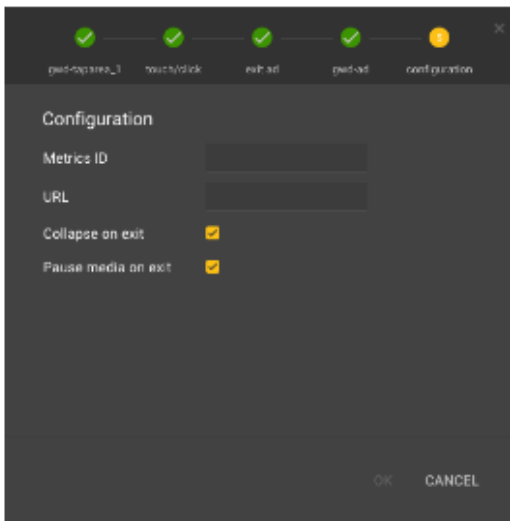
```
<head>
  <meta charset="utf-8" />
  <meta name="ad.size" content="width=1920, height=1080" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <link rel="stylesheet" href="styles/css.css" />
  <script src="scripts/carregalImagensBanner.js"></script>
<!-- Clicktag -->
<script type="text/javascript">
  var clickTag = "http://www.flixmedia.com.br"
</script>
</head>
<body>
<a href="javascript:window.open(clickTag, '_blank'); void(0)"></a>
```

## Evento de saída (click)

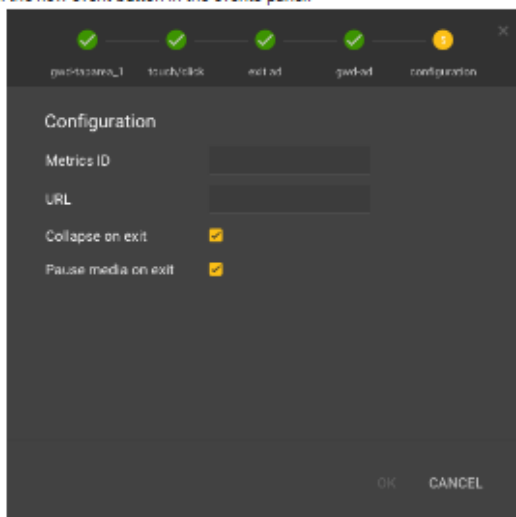
Todos anúncios produzidos no **GoogleWebDesigner** precisam ter um evento de saída (click), conforme procedimentos abaixo:

The final step to creating an effective ad is to add a call-to-action on the expanded page. A typical call-to-action is often a clickable button that takes the user to a company website.

### To add a call-to-action button to your expanded page:



1. Be sure your ad has a button or other clear visual call-to-action element for the user to click or touch.
2. Drag the Tap Area component from the components pane to the stage, position it over the call to action.
3. Click the new event button in the events panel.



4. In the event dialog, select the following options:

Target	The TapArea component (gwd-taparea_1)
Event	Tap Area > Touch/Click
Target	Google Ad > Exit
Receiver	gwd-ad
Configuration	<ul style="list-style-type: none"><li>• <b>Metrics ID</b> - A label, such as <i>clickTag</i> that will make your reporting meaningful.</li><li>• <b>URL</b> - The exit URL.</li><li>• <b>Collapse on exit</b> - Check to close the ad when the user closes the ad</li><li>• <b>Pause media on exit</b> - Check to stop video and audio playback when the user closes the ad</li></ul>

## Creating a dynamic call-to-action with the Tap Area component

The Tap Area component lets you initiate almost any action with a touch or click. One common use is to create a call-to-action. Typically, this will be a static link to a website, but dynamic links are also possible when you're creating a dynamic ad.

The Tap Area component has a dynamic property, **Exit Override URL**, that's accessible through the data binding section of the Dynamic Properties panel. This lets you use a schema that includes individual product URLs, and bind them dynamically to the Tap Area component.

## **Pixels Contadores**

Pixels contadores devem ser enviados separadamente, pois serão adequadamente inseridos no código de forma automatizada.

### **Não recomendamos:**

Usar frameworks javascript como jQuery, Prototype, YUI, Dojo, MooTools, EXTJS e AngularJS pois a probabilidade da página já utilizar um desses recursos é grande e o uso em um anúncio pode comprometer o funcionamento da página ou de outros anúncios.

Não recomendamos o uso de jQuery pois além de comprometer outras funcionalidades da página, serão adicionados 97KB e neste caso pode-se optar por zepto.js que conta com as principais funções e mesma sintaxe, porém com 9.1KB

### **Recomendamos:**

Minifying, existem ferramentas online que fazem a compressão (minifying) de scripts, além disso os principais editores de código também contam com essa funcionalidade.

## **Imagens**

Muitos efeitos (como o gradiente) são possíveis com o uso de CSS. (<http://www.cssmatic.com/gradient-generator> e <http://www.w3schools.com/css/>)

Para uso de vetores as melhores opções são SVG (<http://www.w3schools.com/svg/> e <https://css-tricks.com/using-svg/>) e PNG

O uso de PNG e JPG como 'Save for web do Photoshop', não usam a melhor compressão. Logo experimente outras opções como TinyPNG (<https://tinypng.com/>) e TinyJPG (<https://tinyjpg.com/>)

## **Fontes**

Fontes fora do padrão do sistema acrescentarão diversos KBs ao anúncio, podendo aumentar consideravelmente o tempo de carregamento. Sempre que possível utilize suas fontes como imagens/vetor ou defina-as em classes CSS específicas.

## **CSS**

Nunca use CSS em tags globais como: body, div, span. Isso pode interferir na página e essa boa prática garante que o CSS será aplicado apenas nos elementos corretos do seu anúncio. Coloque CSS in-line ou crie seus próprios IDs ou Classes que não sejam genéricos. Por ex: container, wrapper, header, etc.

## **Vídeo**

Para o uso de vídeo, recomendamos o uso do player do Youtube embedado. Além disso com algumas restrições abaixo:

### **Autoplay:**

Não recomendamos o uso e não permitimos o uso sem polite download.

Browsers como Safari e Opera não aceitam este atributo.

### **Poster:**

É importante adicionar o thumb enquanto o vídeo não é carregado.

### **Controls:**

Caso não desenvolva os próprios controles, este atributo é necessário para que o usuário tenha o controle do vídeo.

Todos os browsers modernos suportam o uso de HTML5:

- Chrome 40+
- Firefox 35+
- Internet Explorer 10+ (incluindo Edge)
- Opera 20+
- Safari 7.0+

É essencial que todos anúncios funcionem nos determinados browsers e versões.

Este site: <https://html5test.com/results/desktop.html> mostra a classificação de cada browser e versão e seu desempenho.