



Especificações para
Produção de peças mobile

➡ Especificações para produção de peças HTML5 mobile

Ao criar uma peça HTML5 e visualizar-la em uma página no seu ambiente tudo tende a funcionar sem problemas, mas é preciso se lembrar que ela será carregada em uma página no site do veículo que possui muito mais elementos e características que uma página HTML “simples” não possui e isso pode influenciar no funcionamento da peça bem como o dispositivo e tamanho de tela utilizado pelo usuário.

A Predicta possui uma série de especificações que têm como objetivo prevenir problemas na sua veiculação, nos preocupamos em garantir que o código de sua peça seja carregado corretamente e sem impactar o site, levando em conta as conexões móveis que muitas vezes são lentas, por isso seguir estas especificações é vital para uma veiculação com sucesso e mínimo esforço possível da sua parte.

Você pode utilizar um dos vários formatos pre-definidos disponibilizados pela Predicta e que possuem especificações simples para produção do material. Estes formatos tornam a produção de peças muito mais simples uma vez que você só precisa criar o material gráfico e não tem que se preocupar com códigos, plugins e outros elementos técnicos.

Se você preferir produzir a sua peça com templates e códigos personalizados, veja abaixo as especificações que a Predicta tem para que sua veiculação possa ser um sucesso. Independente do Adserver ou tecnologia selecionados para veicular suas campanhas, estas são dicas valiosas para qualquer tipo de produção de peças mobile.

O que não é recomendado

➡ O uso de frameworks javascript como jQuery, Prototype, YUI, Dojo, MooTools, EXTJS, AngularJS não é recomendado

Porquê?

- A grande maioria dos veículos já utiliza um destes frameworks em suas páginas, ao incluir eles com sua criação a chance de conflito com o site é muito grande, funcionalidades do site e/ou da sua peça podem parar de funcionar e por isso os veículos não aceitam tags de peças que incluam estes frameworks
- Limite de peso de peça por parte dos veículos. Os veículos consideram como peso da peça todos os elementos que ela carrega, assim, como estas frameworks possuem muito código para abstrair diferentes navegadores, isso as torna muito pesadas, o que aumenta o tempo de carregamento das páginas.

O que sugerimos

Utilizar micro bibliotecas. Você encontra ótimas opções em <http://microjs.com>. Micro bibliotecas oferecem funcionalidades específicas para suas peças, como animações de elementos, criação de desenhos vetoriais (canvas) entre outros, de forma muito mais leve pois não precisam carregar grandes frameworks como jQuery e nem plugins para oferecer o mesmo resultado final. Isso não torna apenas sua peça mais leve, bem como torna ela mais rápida para carregar/executar animações, principalmente em dispositivos com hardware limitado.

As tags mobile da Predicta carregam todos os elementos de forma assíncrona, isso permite que o carregamento inicial da tag seja mais leve facilitando a aprovação pelos veículos

➡ Cuidado com uso de fontes customizadas (web fonts)

🔍 Porquê?

A definição de fontes customizadas pode sobrescrever as fontes padrão do veículo, isso pode gerar problemas no layout do site e consequentemente a peça não ser aprovada pelo veículo. Muitas fontes também são pesadas e grande maioria dos caracteres dela não é usada na criação e isso aumenta o tempo de carregamento para o usuário.

👍 O que sugerimos

Utilize imagens sempre que possível ou defina as suas web fonts com classes css específicas, nunca defina elas em tags globais como body, div, span, p, table, td, etc, pois você irá interferir com as fontes do site. Ao utilizar a definição de fontes em classes e aplicar elas nos elementos desejados você garante que as fontes serão aplicadas apenas nos elementos corretos.

O que não é permitido

➡ **Requisições externas a outros domínios de assets tais como javascript, fontes, imagens, style sheets (css)**

Porquê?

Todos os recursos (javascript, fontes, imagens, css) serão carregados de forma assíncrona pela tag da Predicta para garantir que a peça seja entregue corretamente, requisições externas aumentam o tempo de carregamento e podem até ser reprovadas pelo veículo. Por ser homologada com os veículos, a Predicta hospedará estes recursos e garantirá a entrega da peça da forma correta.

O que sugerimos

Construa sua peça com os assets todos em arquivos separados de forma que possam ser facilmente hospedados pela Predicta. Por exemplo se você possui uma biblioteca ou função javascript de animação para um elemento no código, salve-a em um arquivo .js e insira-a no código HTML usando este arquivo. Outra vantagem é que estes arquivos são mantidos em cache do navegador depois do primeiro carregamento e nas próximas vezes que o usuário for exposto a peça, o carregamento será ainda mais rápido.

Há algumas exceções a este caso como por exemplo, se sua peça utiliza apis públicas como a do Google Maps, ou Twitter que dependem de chaves ou não podem ser baixadas como um arquivo js, podem ser incluídas, mas o veículo precisa ser informado para aprovar ou não a peça por utilizar chamadas externas.

➡ **Nunca associe o carregamento da peça ou do início da animação ao evento “onload” de uma página (carregamento completo do site).**

? Porquê?

A velocidade de carregamento do site influencia no evento de load da página e isso pode fazer com que sua peça nunca seja visualizada a tempo pelo usuário ou o veículo já pode ter outras funções associadas ao carregamento total do site que podem sobrescrever a sua e como o carregamento da tag é feita de forma assíncrona, não necessariamente o início da sua função será feito junto com o carregamento total da página.

👍 O que sugerimos

Se você possuir uma função específica para iniciar uma animação, execute ela no final do código javascript, depois de todos os elementos de script que você carrega.

➡ **Nunca insira todo o código javascript, css no HTML da sua criação**

? Porquê?

O carregamento dos recursos é feito de forma assíncrona pois os veículos possuem limites de peso total da peça e não aceitarão peças acima deste limite, bem como o código JS e CSS junto do HTML dificulta a adaptação das peças para veicular em mobile.

O que sugerimos

Separe os códigos JS e CSS em arquivos, desta forma poderão ser carregados assíncronamente ou com tecnologia polite, permitindo que o veículo aceite a tag. Separar os códigos também simplifica e agiliza a manutenção/modificação destes códigos se necessário sem precisar mexer em todo HTML.

Coloque os arquivos JS e CSS na mesma pasta que o seu HTML, não utilize pastas para estes tipos de arquivo, apenas os referencie no seu HTML por meio das tags script e link acessando-os da mesma pasta onde o HTML com sua peça está.

➡ **Nunca definir estilos CSS em tags globais como *, html, body, div, a, span, img e etc**

Porquê?

Ao definir estilos em tags de forma global, você pode interferir com os estilos já definidos no site do veículo. Isso acarretará em problemas no layout da página e até em como o site irá funcionar no dispositivo do usuário, da mesma forma que estilos do site pre-definidos podem influenciar nos seus elementos, por exemplo, se o site define o tamanho de fonte de uma div de forma global para 12px, isso irá influenciar qualquer elemento div no seu código e o texto dele terá 12px.

O que sugerimos

Defina os estilos dos seus elementos “inline” por meio do atributo “style”, isso previne que seu css seja sobrescrito ou ignorado se a página do veículo

já possuir um css para aquele elemento. Esta também é uma boa forma de garantir que seu estilo será aplicado corretamente.

Outra opção é definir um arquivo css com classes exclusivas para seus elementos e definir estas classes em seus elementos por meio do atributo “class” no elemento

➡ **Nunca defina nomes de variáveis javascript, classes css ou id de elementos com nomes comuns, tais como “animar”, “final”, “botao”, “tocar”, “play”, “btn”, “janela”, “titulo” e etc.**

❓ Porquê?

Embora estes nomes de variáveis javascript, classes css ou id de elementos funcionem perfeitamente em uma página html simples, o seu código e elementos serão carregados em uma página onde já podem existir outros elementos ou códigos com o mesmo nome. Isso irá gerar conflito e certamente seu código não irá funcionar

👍 O que sugerimos

Certifique-se de dar nomes a elementos , variáveis e funções que sejam únicos, de forma que eles não possam ser “iguais” ao da página. Use combinações aleatórias ou nomes bem específicos, por exemplo, ao invés de chamar uma div com id “titulo-animado”, inclua algum dado pertinente a criação/campanha na descrição, altere para “tit_animado_camp_abc”. O mesmo vale para funções javascript, por exemplo uma função chamada “play”, poderia se chamar “chamar_play_camp_abc”, assim a chance de haver o mesmo nome na página é mínimo.

➡ E vídeo?

Peças que possuam vídeo podem ter o vídeo hospedado em sites como YouTube e Vimeo, porém ao utilizar estes serviços você não terá as métricas avançadas de vídeo que a Predicta oferece tais como quantidades de vezes que o usuário tocou o vídeo, percentual assistido, quantas vezes ele voltou ou parou o vídeo. Se você deseja estas métricas deve enviar o seu vídeo no formato mp4 já nas dimensões e peso aceitos pelo veículo.

Tenha em mente que a veiculação em mobile deve sempre ter o foco em desempenho, logo o ideal é que o vídeo tenha o menor tamanho possível no seu peso para que seja carregado o mais rápido possível.