

Ícones e Splashscreen

Transcrição

Nós vimos como fazer a mudança de ícone no *PhoneGap*. Para isto, temos a *tag icon*, que levará o nome do arquivo. No projeto do *PhoneGap*, não demos tanta atenção para esses elementos, mas eles exigem mais atenção, porque cada dispositivo e plataforma tem um tamanho ideal de ícones.

Na página de documentação do [Cordova](https://cordova.apache.org/docs/en/5.0.0/config_ref/images.html), ele irá mostrar como fazer. (https://cordova.apache.org/docs/en/5.0.0/config_ref/images.html).

```
<icon src="res/icon.png"/>
```

Encontraremos a recomendação de que em cada plataforma, deveríamos incluir alguns tamanhos de ícones. Para Android, encontraremos:

```
<platform name="android">
  <icon src="res/android/ldpi.png" density="ldpi" />
  <icon src="res/android/mdpi.png" density="mdpi" />
  <icon src="res/android/hdpi.png" density="hdpi" />
  <icon src="res/android/xhdpi.png" density="xhdpi" />
</platform>
```

Para iOS, teremos uma grande quantidade de ícones, que se referem às versões e dispositivos. Veremos alguns:

```
<platform name="ios">
  <!-- iOS 8.0+ -->
  <!-- iPhone 6 Plus -->
  <icon src="res/ios/icon-60@3x.png" width="180" height="180" />
  <!-- iOS 7.0+ -->
  <!-- iPhone / iPod Touch -->
  <icon src="res/ios/icon-60.png" width="60" height="60" />
  <icon src="res/ios/icon-60@2x.png" width="120" height="120" />
</platform>
```

Imagina se tivéssemos que criar estes elementos manualmente. Temos algumas ferramentas que facilitam a criação dos mesmos, uma delas é o ***Ionic***. Trata-se de um *framework* bastante completo para se trabalhar com aplicativos híbridos no Cordova, que permite criar componente, telas, além de alguns serviços opcionais. Entre eles, o ***Ionic Resources***, que permite gerar os ícones nos tamanhos corretos, válidos para cada plataforma utilizada. Não focaremos no *Ionic*, mas iremos conhecer uma característica do serviço.

Primeiramente, iremos instalar o *Ionic*. Ele é um módulo do *Node* e por isso, usaremos o comando `npm install ionic`. Como queremos instalá-lo globalmente, usaremos o `-g`. E para darmos permissão de administrador tanto no Mac como no Linux, adicionaremos `sudo`.

```
garconapp $ sudo npm install -g ionic
```

Eu já o tenho instalado. Com o comando `ionic resources`, poderemos gerar todos os ícones necessários. No nosso caso, iremos criar um ícone grande e depois, o *Ionic* irá redimensioná-lo em diversos tamanhos. Usaremos o arquivo ícone com o logo do "Só de Cenoura", que tem a dimensão de 1024x1024. Os ícones terão tamanhos menores.

No projeto do Garçonapp criaremos um diretório novo, chamado `resources`, é onde geralmente colocamos os ícones e as *Splash screens* no Cordova, incluindo o arquivo `icon.png`. Ele receberá o nome `icon` e precisará ter uma forma quadrada, ocupando a tela inteira. Não é recomendável fazer um ícone redondo, porque existem diferenças nas plataformas que poderão alterar a visualização do ícone. No iOS, é a própria plataforma que cria o ícone redondo. Também colocaremos no diretório o arquivo `splash.png`. Ele possui uma dimensão 4096x4096, bem maior do que iremos precisar. A imagem será quadrada, porque a partir dela, ele irá recortar em outros tamanhos - tanto no modo retrato como em paisagem.

Voltaremos para a linha de comando, com o *Ionic* instalado, rodaremos `ionic resources`. Ele irá subir os arquivos de imagens para o servidor do Ionic, que possui um serviço gratuito de geração de ícones com as dimensões necessárias para Android e iOS.

```
garconapp $ ionic resources
Ionic icon and splash screen resources generator
uploading icon.png
uploading splash.png
icon.png (1024x1024) upload complete
splash.png (4096x4096) upload complete
generating splash android drawable-port-xxxhdpi-screen.png (1280x1920)...
generating splash android drawable-port-xxhdpi-screen.png (960x1600)...
generating splash android drawable-port-xhdpi-screen.png (720x1280)...
splash android drawable-port-xhdpi-screen.png (720x1280) generated
generating splash android drawable-port-hdpi-screen.png (480x800)...
/...
```

O processo irá demorar um pouco, mas podemos acompanhá-lo. Ele irá gerar duas pastas novas no diretório `resources`, uma chamada `android` e outra `ios`. Se clicarmos nas pastas, veremos ícones e *Splash screens* de diversos tamanhos, baseados nas imagens originais. Como especificamos que o app funcionará na orientação `portrait` ("retrato"), ele criou *Splash screens* apenas neste formato.

Você pode perguntar: precisarei incluir todos os arquivos no `xml`? Não, o próprio *Ionic*, por ser baseado no Cordova, gerou os ícones necessário no `config.xml`. Inclusive, inclui os ícones dentro da `tag platform`, onde havíamos adicionado a preferência específica para iOS.

```
<platform name="ios">
  <preference name="DisallowOverscroll" value="true" />
  <icon src="resources/ios/icon/icon.png" width="57" height="57" />
  <icon src="resources/ios/icon/icon@2x.png" width="114" height="114" />
  <icon src="resources/ios/icon/icon-40.png" width="40" height="40" />
  <icon src="resources/ios/icon/icon-40@2x.png" width="80" height="80" />
/...
</platform>
```

Ele também já adicionou diversos tamanhos da *Splash screens*:

```
<splash src="resources/ios/splash/Default-568h@2x~iphone.png" width="640" height="1136" />
<splash src="resources/ios/splash/Default-667h.png" width="750" height="1334" />
```

```
<splash src="resources/ios/splash/Default-736h.png" width="1242" height="2208" />
/...
```

Para os ícones Android, ele fez o mesmo processo: criou o `platform` específico e depois, todo o `xml` necessário.

```
<platform name="android">
  <preference name="DisallowOverscroll" value="true" />
  <icon src="resources/android/icon/drawable-ldpi-icon.png" density="ldpi"/>
  <icon src="resources/android/icon/drawable-mdpi-icon.png" density="mdpi"/>
  <icon src="resources/android/icon/drawable-hdpi-icon.png" density="hdpi"/>

/...
</platform>
```

Após gerarmos tais componentes, iremos rodar o app novamente. Veremos que o ícone utilizado na Home será o recém criado, assim como a *Splash screen*. Logo, quando ele carregar o app, veremos a lista de produtos do cardápio.

Se você desejar exportar manualmente cada um dos ícones, poderá incluí-los no editor de texto. Mas, um serviço como o Ionic facilita o trabalho.

Temos a opção de criar ícones e *Splash screens* específicos para cada plataforma. Para isto, no diretório `resource`, nas pastas que serão criadas automaticamente com os nomes `ios` e `android`, iremos incluir os arquivos de base - mas que necessariamente receberão os nomes `icon.png` e `splash.png`. Ou seja, além de termos um `png` na raiz, poderemos ter uma imagem dentro de cada plataforma. Assim, a partir dela, serão gerados os ícones específicos. No caso, usaremos apenas o arquivo na raiz do diretório. Iremos praticar o conteúdo nos exercícios.