

## Como configurar o PhoneGap

### Transcrição

Nós terminamos de utilizar o *PhoneGap Build* para instalar o app no Android, como um aplicativo normal. Porém, alguns assuntos ficaram em aberto, como o nome e o ícone.

No site do *PhoneGap Build*, vemos que ele gerou um nome padrão, "PG Build App", e inseriu um ícone esquisito. Nós podemos configurar estes elementos. Veremos como fazer tais ajustes.

No terminal, abriremos o projeto, em que temos HTML e o CSS. Criaremos um arquivo novo chamado `config.xml`, com algumas configurações para o nosso aplicativo.

Vamos criar o arquivo `xml`, que irá configurar o início do arquivo. Para isto, ele seguirá a especificação do WDC, com a tag `widget` - que receberá algumas configurações como o nome ( `name` ).

```
<?xml version="1.0" encoding="UTF-8" ?>
<widget>
  <name>Só de Cenoura</name>

</widget>
```

Quando trabalhamos com um `xml` precisamos indicar os cabeçalhos, com os *namespaces*. Iremos adicionar o `xmlns` s com o endereço do `http://www.w3.org/ns/widgets` .

```
<?xml version="1.0" encoding="UTF-8" ?>
<widget xmlns="http://www.w3.org/ns/widgets">

  <name>Só de Cenoura</name>

</widget>
```

Também vamos passar outro *namespace*, chamado `gap`, onde posso colocar algumas extensões específicas do `phonegap` .

```
<?xml version="1.0" encoding="UTF-8" ?>
<widget xmlns="http://www.w3.org/ns/widgets" xmlns:gap="http://phonegap.com/ns/1.0"
  >

  <name>Só de Cenoura</name>

</widget>
```

Podemos colocar os *namespaces* na mesma linha.

Existem outros dois parâmetros importantes que precisamos incluir. O primeiro é a versão ( `version` ) do app.

Para termos o controle da quantidade de versões do aplicativo *PG Build App*, vamos especificar o número do app no código. Neste caso, iremos usar a versão "1.0.0", mas podemos ir implementando outros números como "1.0.1" ou "1.0.2". Outro atributo importante é `id`, que irá configurar qual será o nome do pacote do app. Esse pacote deve ser igual para todos os aplicativos. Geralmente, utilizamos o nome do pacote ao contrário. Por exemplo, o meu site chama `sergiolopes.org`, logo usarei `org.sergiolopes` e depois acrescentaremos nome do app `cenouraapp`.

```
<?xml version="1.0" encoding="UTF-8" ?>
<widget xmlns="http://www.w3.org/ns/widgets" xmlns:gap="http://phonegap.com/ns/1.0" version="1.0"
  >

  <name>Só de Cenoura</name>

</widget>
```

Sugerimos que quando você for fazer o exercício, utilize um `id` diferente, baseado no seu próprio domínio. Caso contrário, se você tentar subir seu aplicativo para um loja, ele não será aceito.

Após configurado o cabeçalho, já sabemos como trabalhar com o `name`, podemos incluir outras *tags*. Vamos trabalhar com a `description` que permitirá incluir descrições maiores na configuração do aparelho.

```
<description>
  Menu do restaurante Só de Cenoura
</description>
```

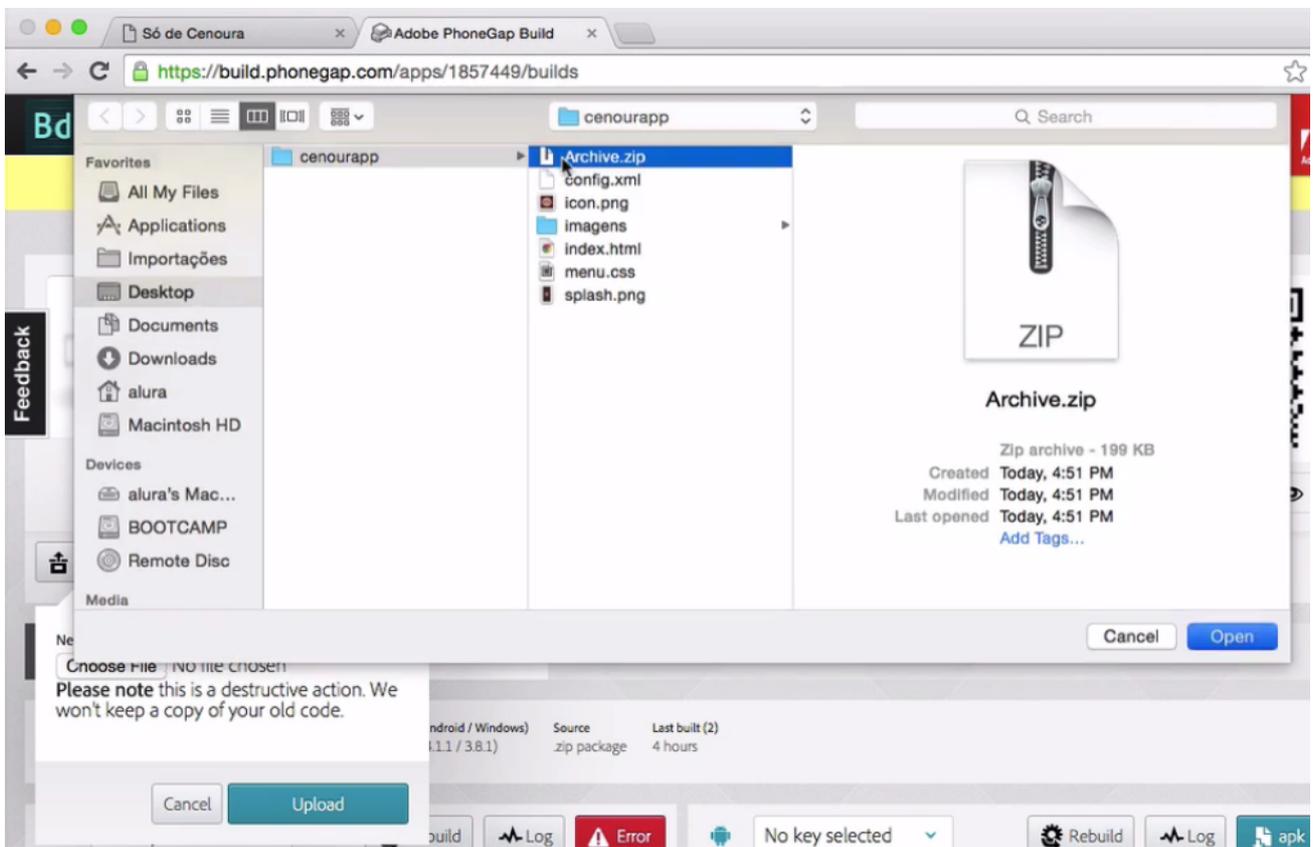
Podemos incluir o nome do autor do app.

```
<author href="http://sergiolopes.org" email="qualquer@sergiolopes.org">
  Sérgio Lopes
</author>
```

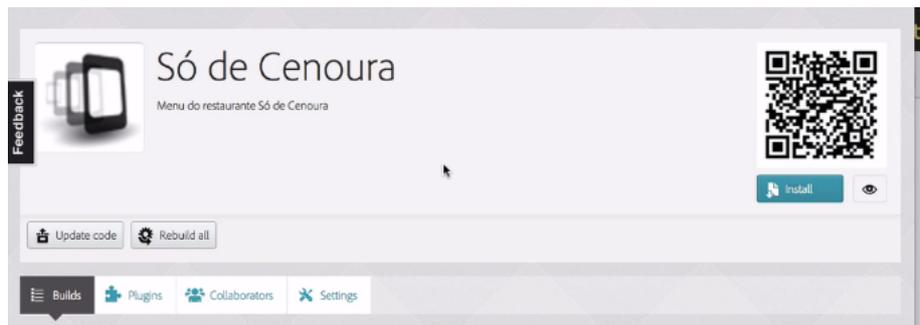
Meu `xml` ficou na raiz do projeto, junto com o `HTML`.

Para subir o projeto no *PhoneGap Build*, iremos zipar o arquivo novamente, mas incluindo o arquivo `config.xml`.

Voltaremos para o site do *PhoneGap Build*, clicaremos em *Update code*, depois em *Choose File* e selecionaremos o arquivo para o *upload*.



Nosso arquivo irá aparecer já com o nome e a descrição alterados.



Se fizéssemos a instalação do aplicativo, ele teria o nome novo.

Em seguida iremos configurar o ícone. No projeto, temos um arquivo chamado `icon.png`, que tem uma imagem do "Só de Cenoura".



Este será o ícone que estará na Home do usuário. Temos também outro arquivo, chamado `splash.png`, que é a tela de carregamento (*splash screen*).



A nossa imagem foi criada com as dimensões para Android.

Para usar o ícone, precisamos voltar na aba XML e utilizar uma configuração apontando a imagem explicitamente. Criaremos a tag `icon`, que irá receber o `source`. Acrescentaremos também o nome do arquivo.

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

Para trabalhar com a *splashscreen*, usaremos a tag do *PhoneGap*, `gap:splash`, com o nome do arquivo.

```
<plugin name="cordova-plugin-splashscreen" />  
<splash src="splash.png" />
```

Nosso `xml` está assim:

```
<?xml version="1.0" encoding="UTF-8" ?>  
<widget xmlns="http://www.w3.org/ns/widgets" xmlns:gap="http://phonegap.com/ns/1.0" version="1.0"  
>  
  
  <name>Só de Cenoura</name>  
  
  <description>  
    Menu do restaurante Só de Cenoura  
  </description>
```

```
<author href="http://sergiolopes.org"
      email="qualquer@sergiolopes.org"> Sérgio Lopes
</author>

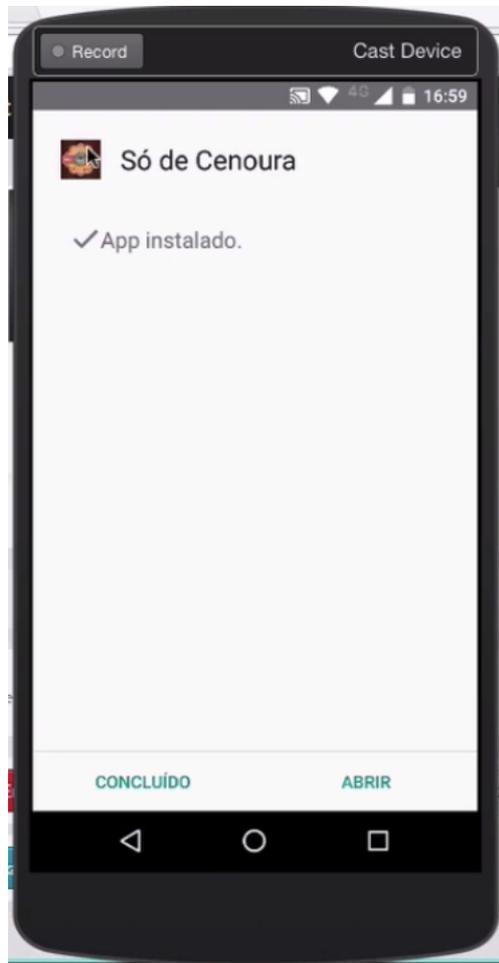
<icon src="icon.png" />
<plugin name="cordova-plugin-splashscreen" />
<splash src="splash.png" />

</widget>
```

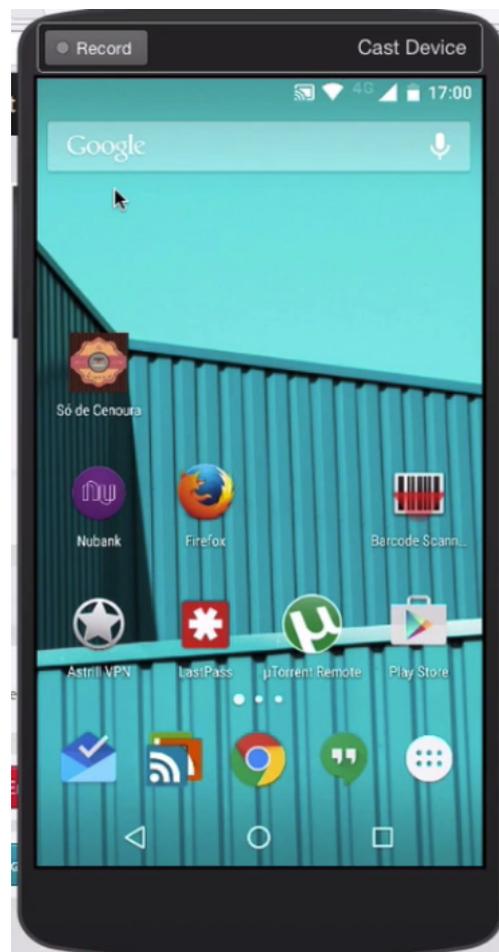
Vamos testar o novo código. Vamos apagar o arquivo .zip anterior e criar outro com as alterações feitas no `xml`. Depois, faremos o *upload* da versão mais recente do *PhoneGap Build*. Com a nova configuração, agora aparece o ícone do "Só de Cenoura".



Se baixarmos a última versão do arquivo `.apk` no celular, também veremos o ícone na Home. Repetiremos o processo de instalação feito antes, com o QR Code. Ele irá abrir o navegador e irá baixar o arquivo. Iremos instalá-lo e já poderemos visualizar o ícone.



Qual é a grande diferença agora? Se formos no menu, o novo ícone já estará disponível para colocarmos na Home.



Ao clicarmos no ícone, veremos que o aplicativo funciona corretamente.

Ainda temos outras configurações adicionais que podem ser feitas no `.xml`, como a `tag` `preference`. Ela permite criar especificações do app, por exemplo, que o usuário não rotacione a página e utilize apenas o modo retrato. Iremos incluir uma `name="orientation"` com o valor igual a retrato (`portrait`).

```
<preference name="orientation" value="portrait" />
```

Se "buildarmos" novamente o app, o usuário não poderá mais rotacionar a visualização. Podemos criar outra preferência para que o aplicativo seja `fullscreen` e ocupe a tela completa - o que pode ser uma opção interessante para um cardápio.

```
<preference name="orientation" value="portrait" />
<preference name="fullscreen" value="true" />
```

Temos outras configurações, para alterar as cores, por exemplo. Ainda que o *PhoneGap Build* nos ofereça a opção de "buildar" o aplicativo para outras plataformas, podemos especificar no código que ele funcionará apenas para *Android*.

```
<platform name="android" />
```

Ele irá "buildar" apenas para plataforma *Android*

Você encontrará mais detalhes sobre preferências, nos [exercícios \(https://cursos.alura.com.br/course/cordova-phonegap/task/14251\)](https://cursos.alura.com.br/course/cordova-phonegap/task/14251).

Agora vamos subir a nova versão do arquivo no *PhoneGap Build*. Observe que na plataforma, ele só irá permitir "buildar" para Android. Ele seguiu corretamente o que configuramos no `xml`.

App ID	Version	Created by	PhoneGap (iOS / Android / Windows)	Source	Last built (6)
1857449	1.0.0	slopes@gmail.com	cli-5.2.0 (3.9.1 / 4.1.1 / 3.8.1)	zip package	9 minutes

Vamos repetir novamente o processo de instalação com o QR Code, ele baixará a versão mais recente do aplicativo. Após a instalação ser finalizada, iremos abrir o aplicativo. Agora, o aplicativo estará *fullscreen*.



Não podemos visualizar o botão de voltar, nem a barra de status. Essa preferência não é recomendável para qualquer tipo de app, mas no nosso caso, queremos oferecer uma experiência imersiva. Por exemplo, o restaurante disponibilizará *tablets* que serão usados simplesmente como cardápios de pedidos. Não é necessário que o usuário saia do aplicativo. Também, não será possível rotacionar o aplicativo. Neste caso, trabalhar com estas preferências pode ser interessante.

Nos exercícios, você terá disponível outras opções de configurações que podem ser aplicadas.