

Programação Móvel

Introdução ao Ionic

Prof. Dr. Diego R. Antunes

drantunes@utfpr.edu.br

Departamento de Computação
Universidade Tecnológica Federal do Paraná

Ambiente de Desenvolvimento

Tutorial

O tutorial foi atualizado em relação às configurações de proxy para uso do ambiente de desenvolvimento na UTFPR. Também há instruções para remover estas configurações para fora da UTFPR.

Proxy

As configurações de proxy são necessárias para o gerenciador de pacotes do Node, Atom e Github.

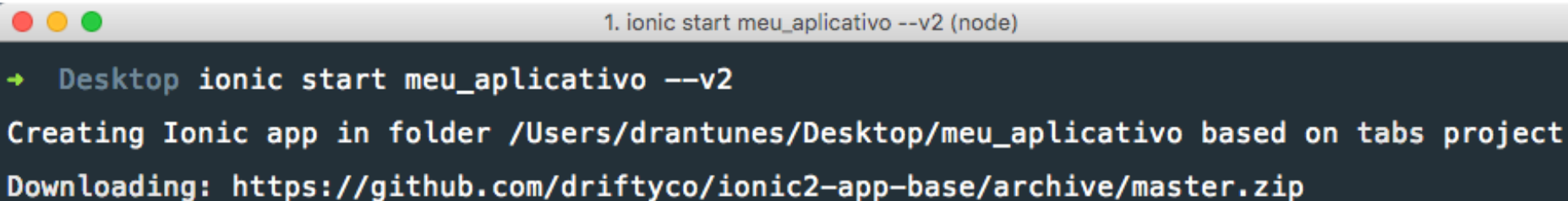
Novo Aplicativo Ionic

Criando um aplicativo Ionic

- 1 *Abra o Terminal ou CMD (Windows)*
- 2 *Entre na pasta que você deseja salvar o projeto*
- 3 *Execute: **ionic start** <nome_do_projeto> --v2*

Criando um aplicativo Ionic

Ao executar o comando `ionic start`, o CLI irá copiar uma aplicação modelo do Github e instalar no diretório que você escolheu.



```
1. ionic start meu_aplicativo --v2 (node)

→ Desktop ionic start meu_aplicativo --v2
Creating Ionic app in folder /Users/drantunes/Desktop/meu_aplicativo based on tabs project
Downloading: https://github.com/driftyco/ionic2-app-base/archive/master.zip
```

Criando um aplicativo Ionic

*Em seguida, os pacotes **npm** serão instalados em seu repositório na pasta **node_modules**. Dica: Ao salvar seu projeto como zip, não salve a pasta **node_modules**.*

```
→ Desktop ion start meu_aplicativo --v2
Creating Ionic app in folder /Users/drantunes/Desktop/meu_aplicativo based on tabs project
Downloading: https://github.com/driftyco/ionic2-app-base/archive/master.zip
[=====] 100% 0.0s
Downloading: https://github.com/driftyco/ionic2-starter-tabs/archive/master.zip
[=====] 100% 0.0s
Installing npm packages...
```


Criando um aplicativo Ionic

*Então, em outro computador para instalar os pacotes npm basta você entrar na pasta do projeto e executar o comando **npm install** (os pacotes serão instalados).*

```
→ Desktop ion start meu_aplicativo --v2
Creating Ionic app in folder /Users/drantunes/Desktop/meu_aplicativo based on tabs project
Downloading: https://github.com/driftyco/ionic2-app-base/archive/master.zip
[=====] 100% 0.0s
Downloading: https://github.com/driftyco/ionic2-starter-tabs/archive/master.zip
[=====] 100% 0.0s
Installing npm packages...
```

Criando um aplicativo Ionic

Ao final da instalação o CLI mostra algumas informações principais.

```
Adding in iOS application by default
```

```
add to body class: platform-ios
```

```
Saving your Ionic app state of platforms and plugins
```

```
Saved platform
```

```
Saved plugins
```

```
Saved package.json
```

```
🎶 🎶 🎶 🎶 Your Ionic app is ready to go! 🎶 🎶 🎶 🎶
```

Criando um aplicativo Ionic

Por padrão, o ionic define a plataforma para IOS. Então devemos alterar isso para android no momento de testar.

```
Adding in iOS application by default
```

```
add to body class: platform-ios
```

```
Saving your Ionic app state of platforms and plugins
```

```
Saved platform
```

```
Saved plugins
```

```
Saved package.json
```

```
🎵 🎵 🎵 🎵 Your Ionic app is ready to go! 🎵 🎵 🎵 🎵
```

Criando um aplicativo Ionic

Make sure to cd into your new app directory:

```
cd meu_aplicativo
```

To run your app in the browser (great for initial development):

```
ionic serve
```

To run on iOS:

```
ionic run ios
```

To run on Android:

```
ionic run android
```

Criando um aplicativo Ionic

Make sure to cd into your new app directory:

```
cd meu_aplicativo
```

To run your app in the browser (great for initial development):

```
ionic serve
```

To run on iOS:

```
ionic run ios
```

To run on Android:

```
ionic run android
```

Criando um aplicativo Ionic

Make sure to cd into your new app directory:

```
cd meu_aplicativo
```

To run your app in the browser (great for initial development):

```
ionic serve
```

To run on iOS:

```
ionic run ios
```

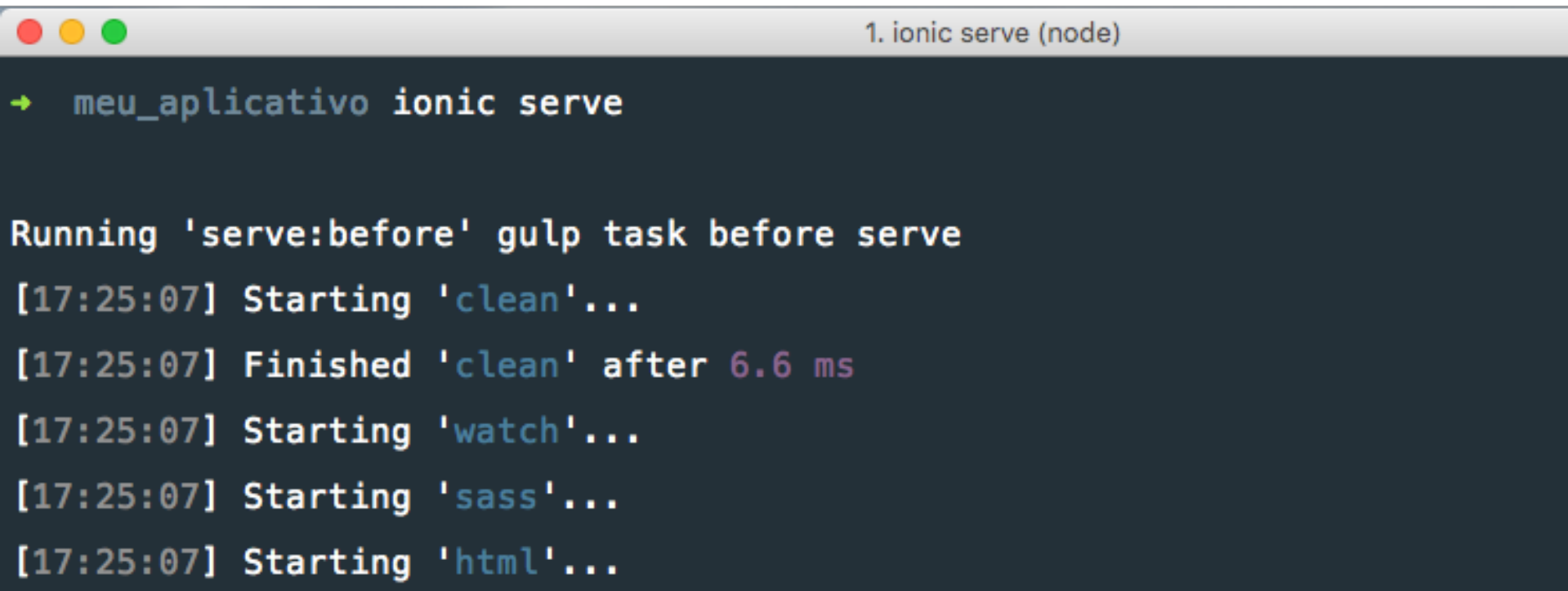
To run on Android:

```
ionic run android
```

Testando no Browser

Testando o App no Browser

Ao executar o comando *ionic serve*, os arquivos serão compilados e a aplicação será carregada no navegador.



```
1. ionic serve (node)

→ meu_aplicativo ionic serve

Running 'serve:before' gulp task before serve
[17:25:07] Starting 'clean'...
[17:25:07] Finished 'clean' after 6.6 ms
[17:25:07] Starting 'watch'...
[17:25:07] Starting 'sass'...
[17:25:07] Starting 'html'...
```


Testando o App no Browser

Você fará isso somente uma vez, quando iniciar o desenvolvimento do aplicativo.

```
Running live reload server: http://localhost:35729
```

```
Watching: www/**/*, !www/lib/**/*
```

```
✓ Running dev server: http://localhost:8100
```

```
Ionic server commands, enter:
```

```
restart or r to restart the client app from the root
```

```
goto or g and a url to have the app navigate to the given url
```

Testando o App no Browser

Em seguida, você irá apenas desenvolver seu aplicativo e a cada vez que salvar, o ionic compila e atualiza para você automaticamente.

```
Running live reload server: http://localhost:35729
```

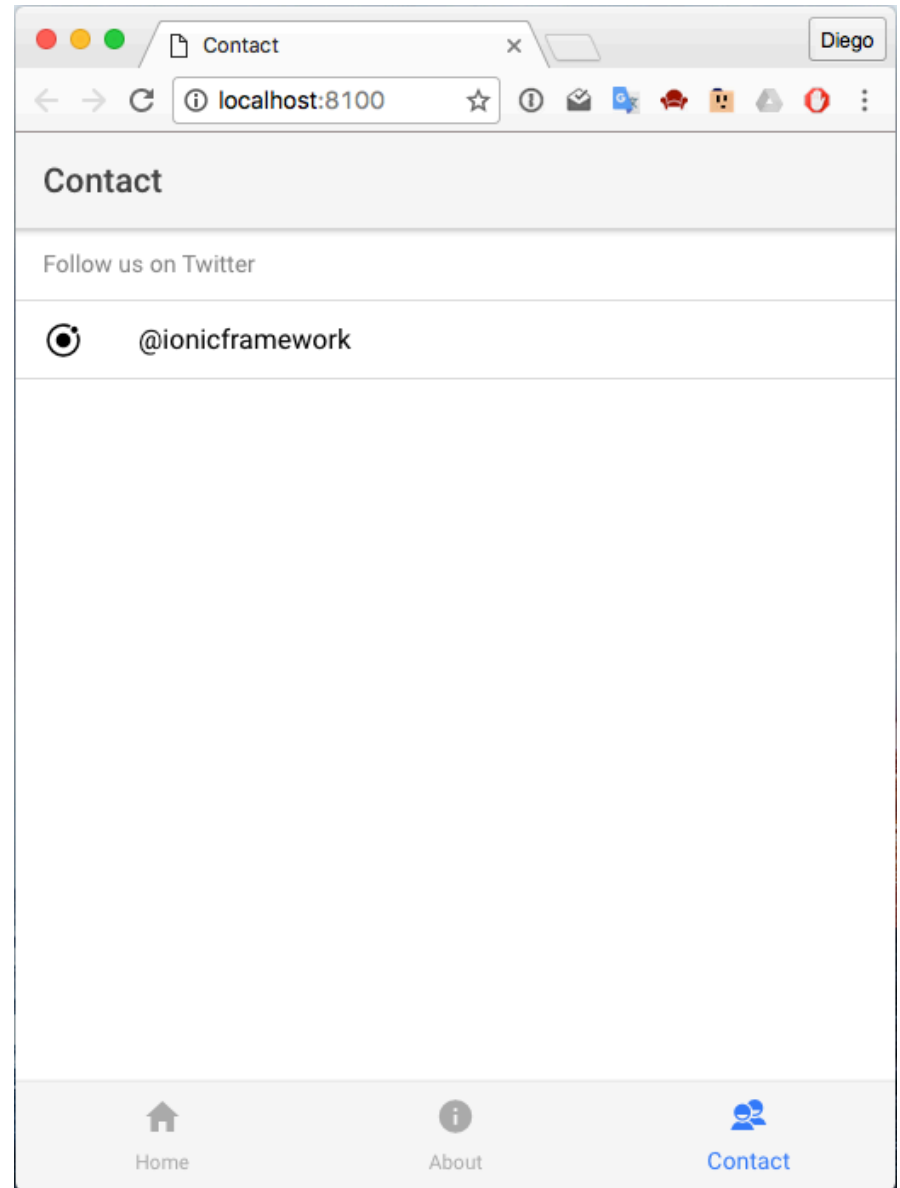
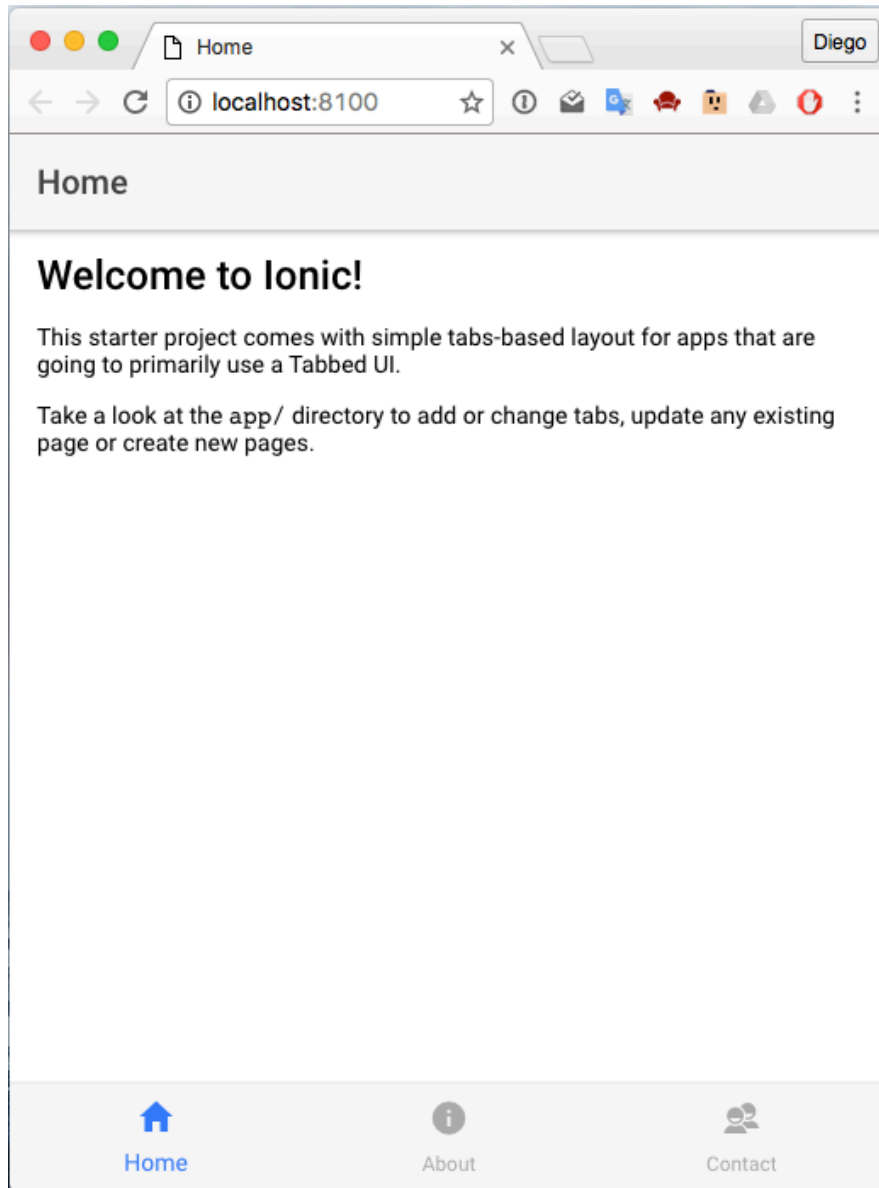
```
Watching: www/**/*, !www/lib/**/*
```

```
✓ Running dev server: http://localhost:8100
```

```
Ionic server commands, enter:
```

```
restart or r to restart the client app from the root
```

```
goto or g and a url to have the app navigate to the given url
```



Testando outras plataformas

No navegador você pode digitar uma URL especial (localhost:8100/ionic-lab) que aciona o Ionic Lab, que mostra o aplicativo com a Interface de cada plataforma: Android, iOS, Windows.



Home

Welcome to Ionic!

This starter project comes with simple tabs-based layout for apps that are going to primarily use a Tabbed UI.

Take a look at the `app/` directory to add or change tabs, update any existing page or create new pages.

Home About Contact

Android

Home

Welcome to Ionic!

This starter project comes with simple tabs-based layout for apps that are going to primarily use a Tabbed UI.

Take a look at the `app/` directory to add or change tabs, update any existing page or create new pages.

Home About Contact

Windows

HOME

Home About Contact

Welcome to Ionic!

This starter project comes with simple tabs-based layout for apps that are going to primarily use a Tabbed UI.

Take a look at the `app/` directory to add or change tabs, update any existing page or create new pages.

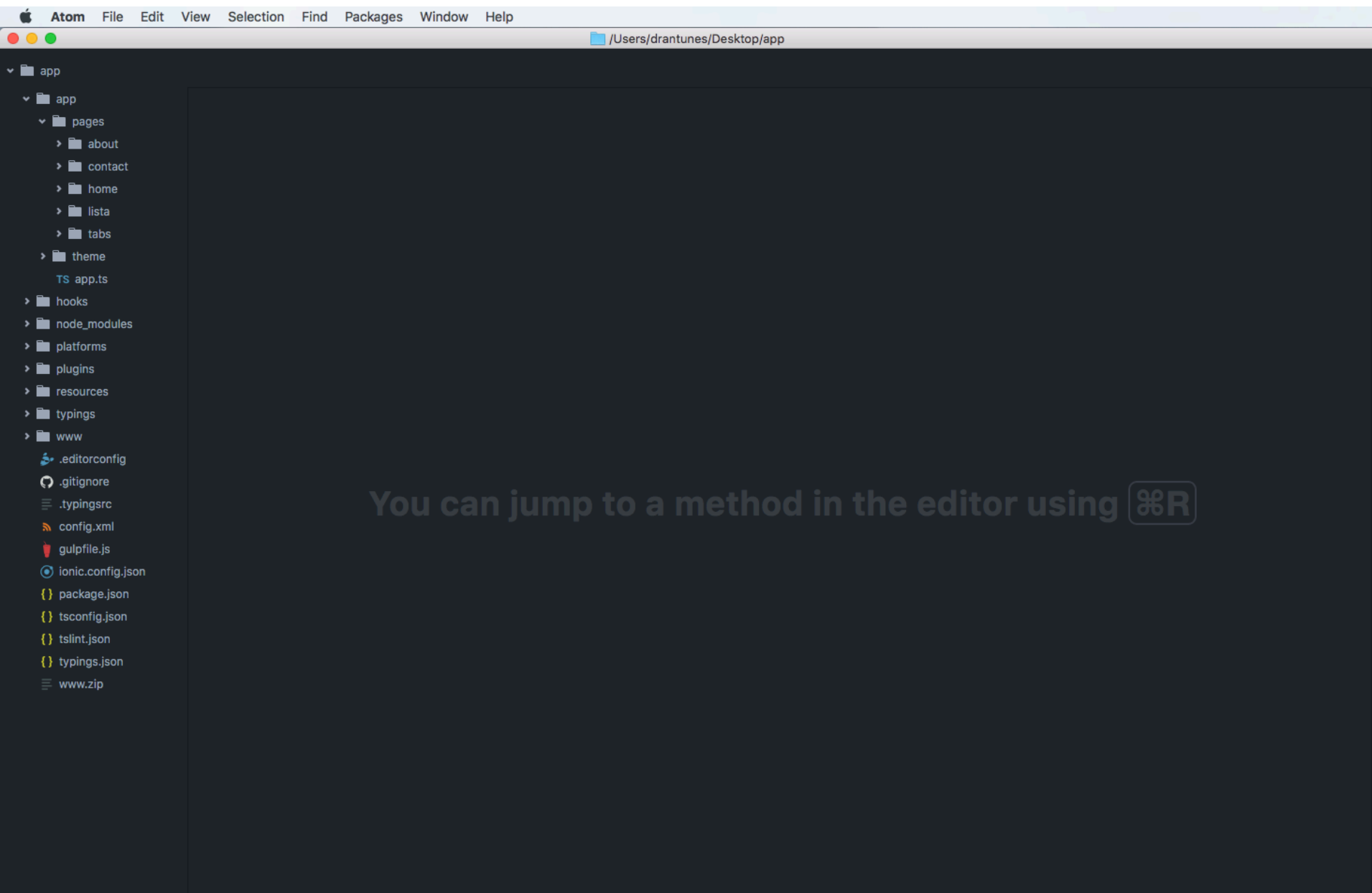
Abrindo o projeto no Atom

*No Atom (IDE), é possível utilizar o plugin **ionic preview** (está no tutorial de instalação). Desta maneira, você pode visualizar seu aplicativo dentro da própria IDE Atom.*

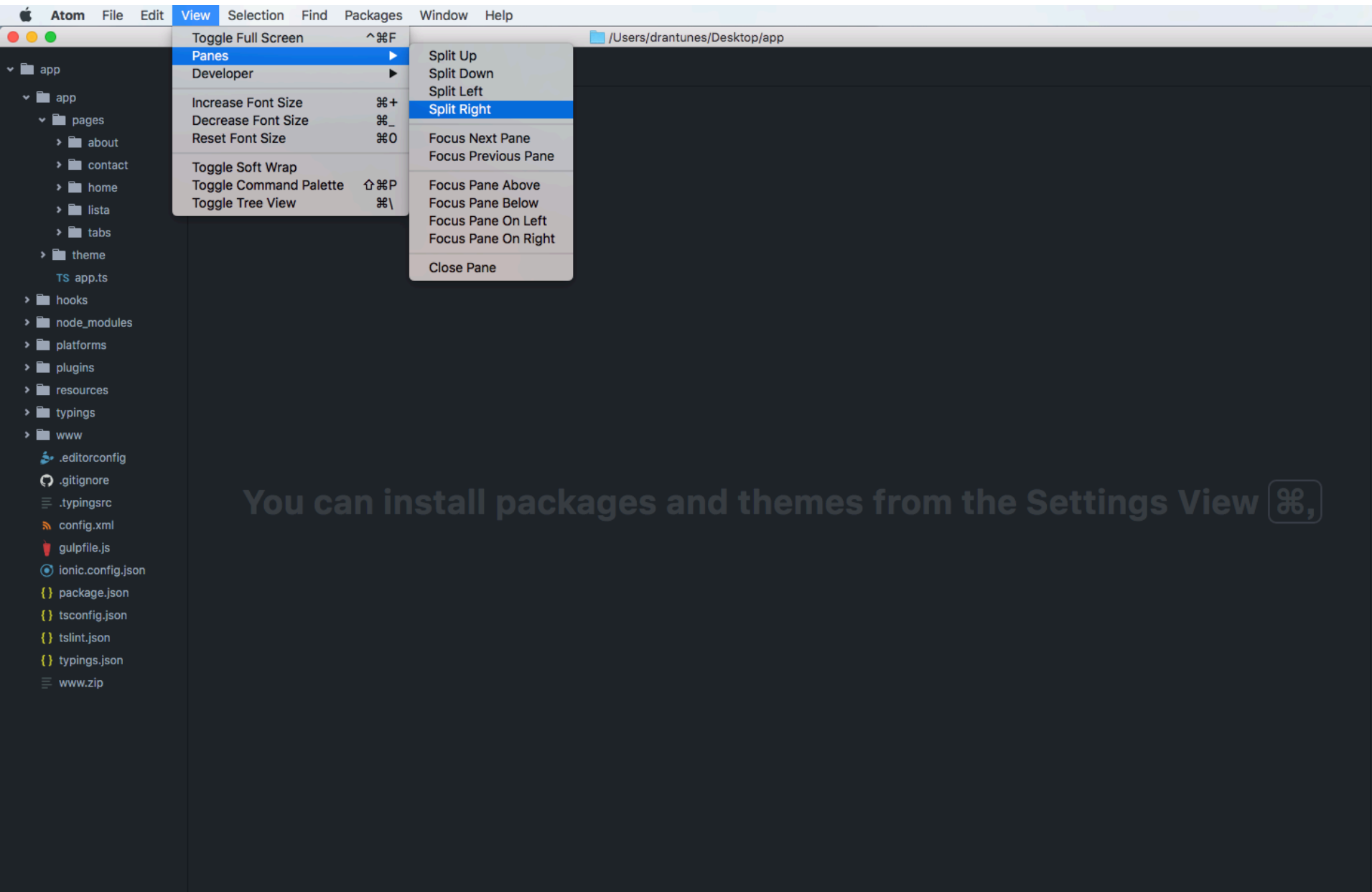
Abrindo o projeto no Atom

Primeiro, execute o comando `ionic serve -b` no Terminal ou CMD para iniciar o ionic CLI. Em seguida ...

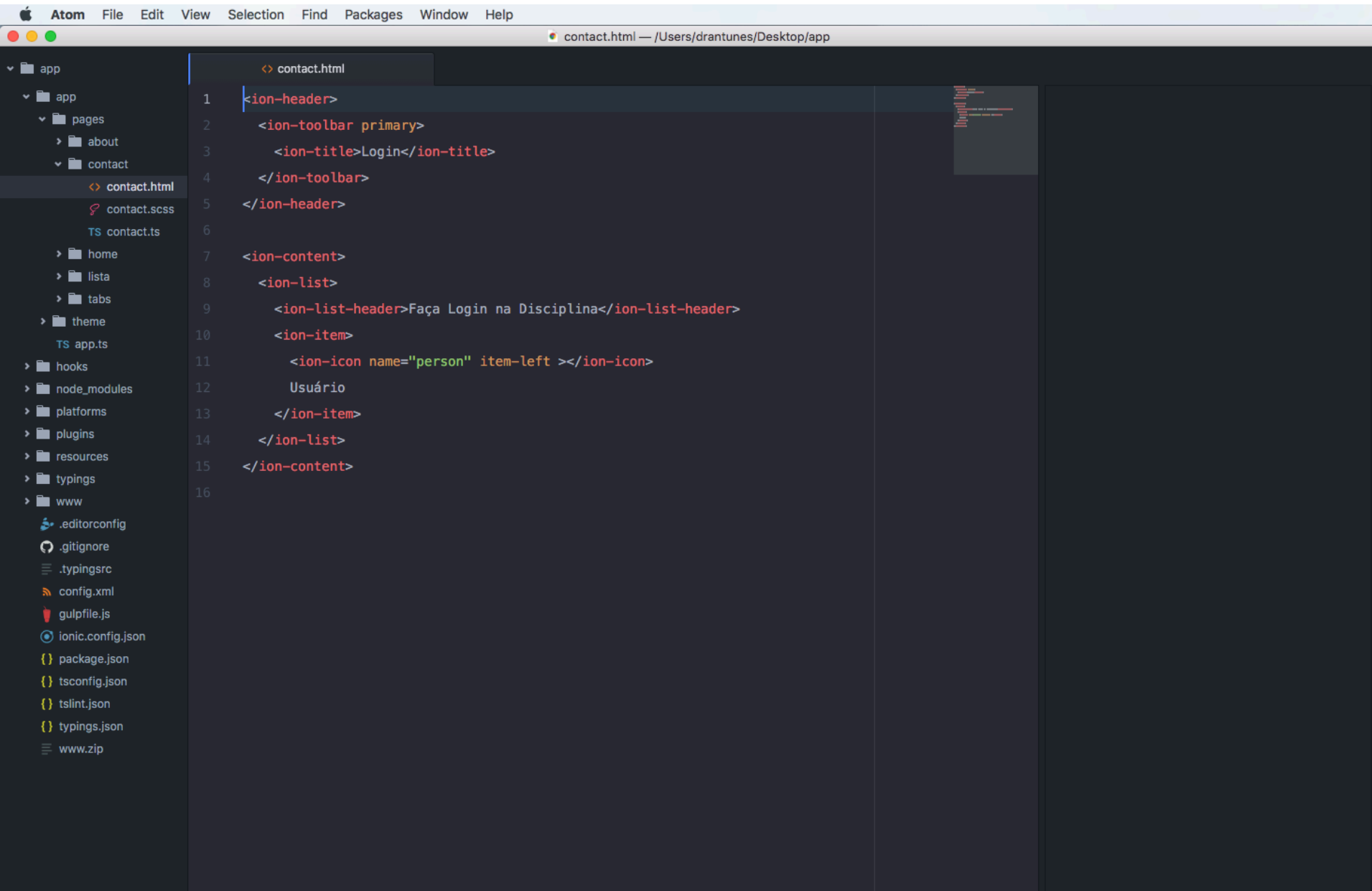
Abra o projeto no Atom



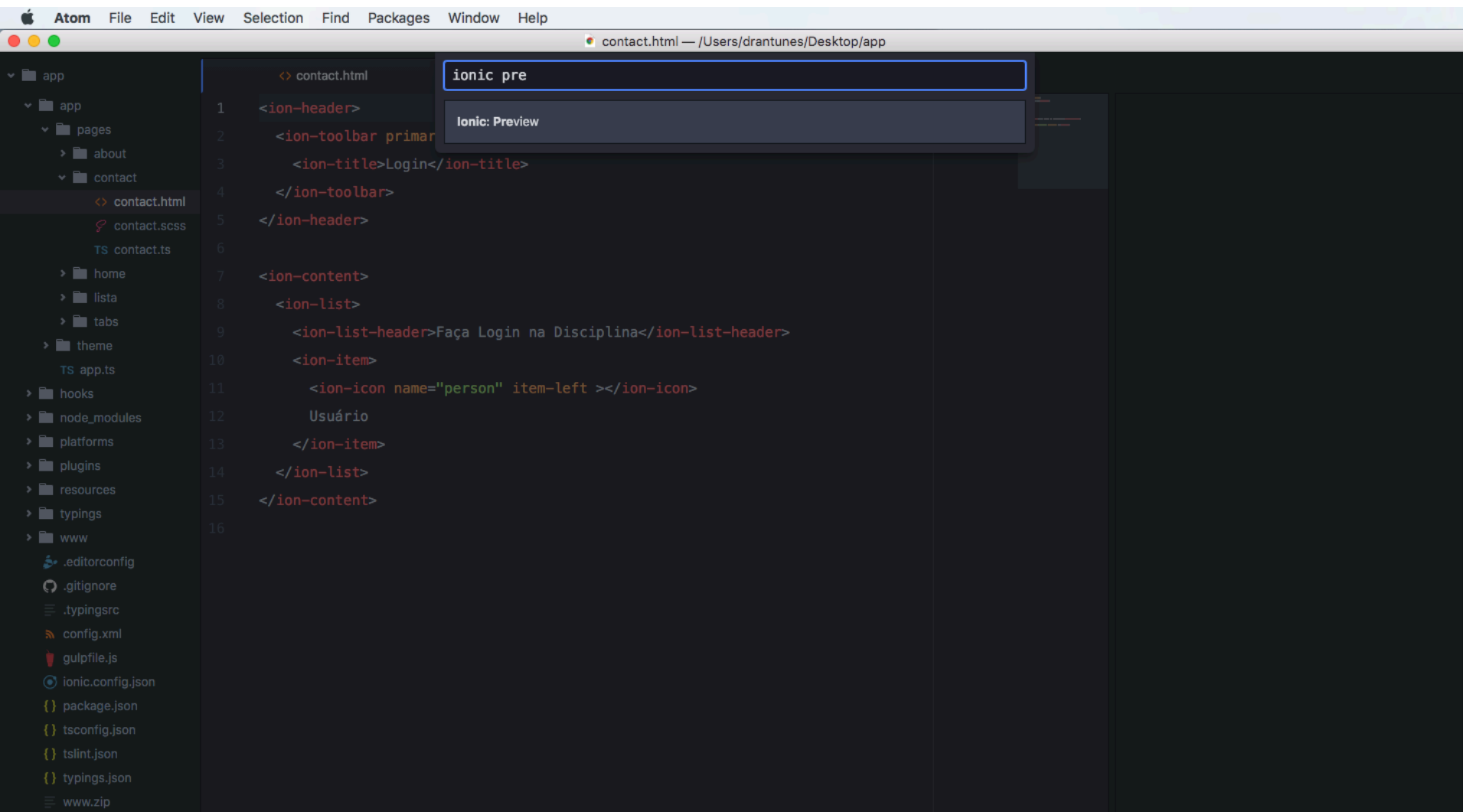
Em View, faça um Split da tela do Atom



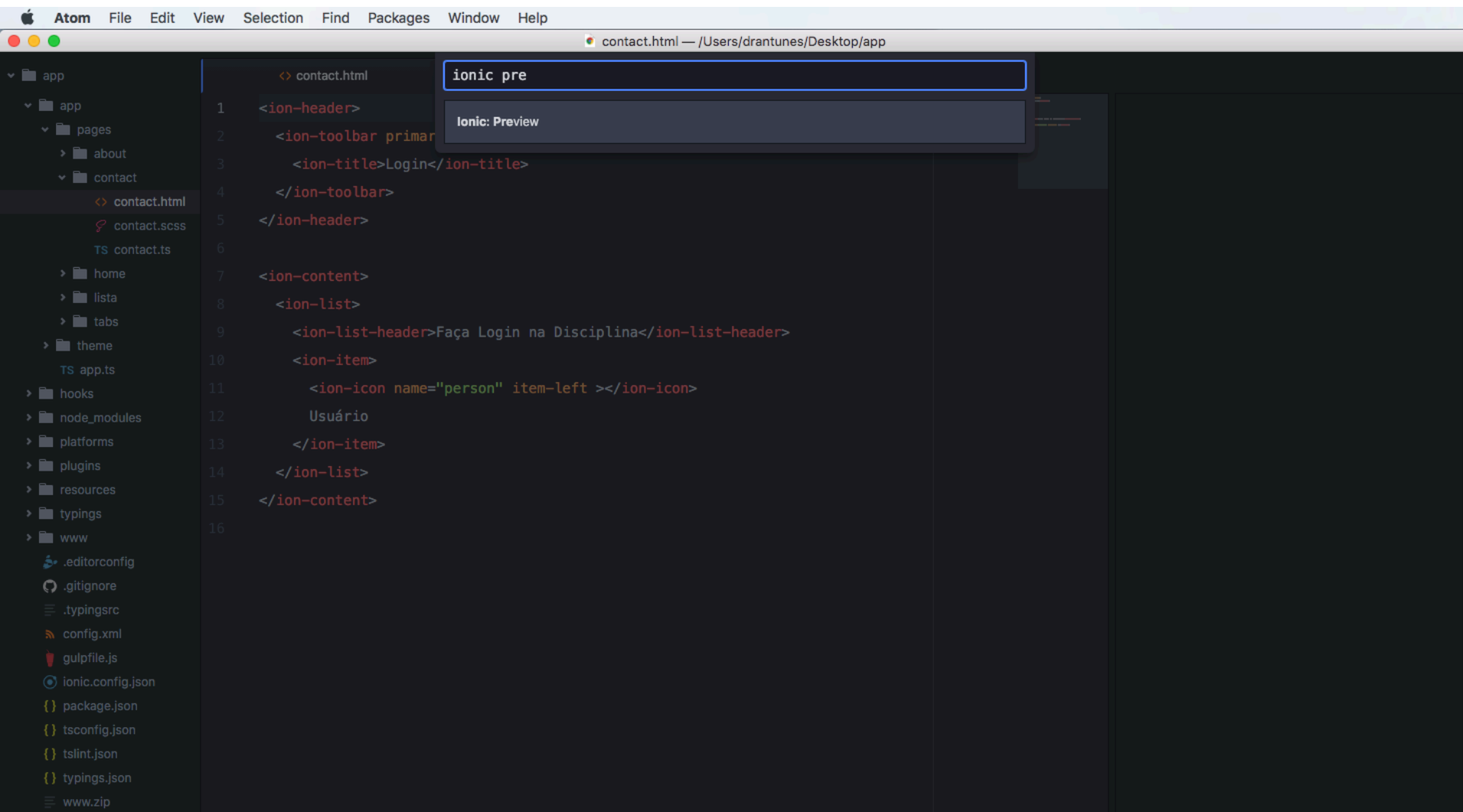
Então você abrir o código na esquerda



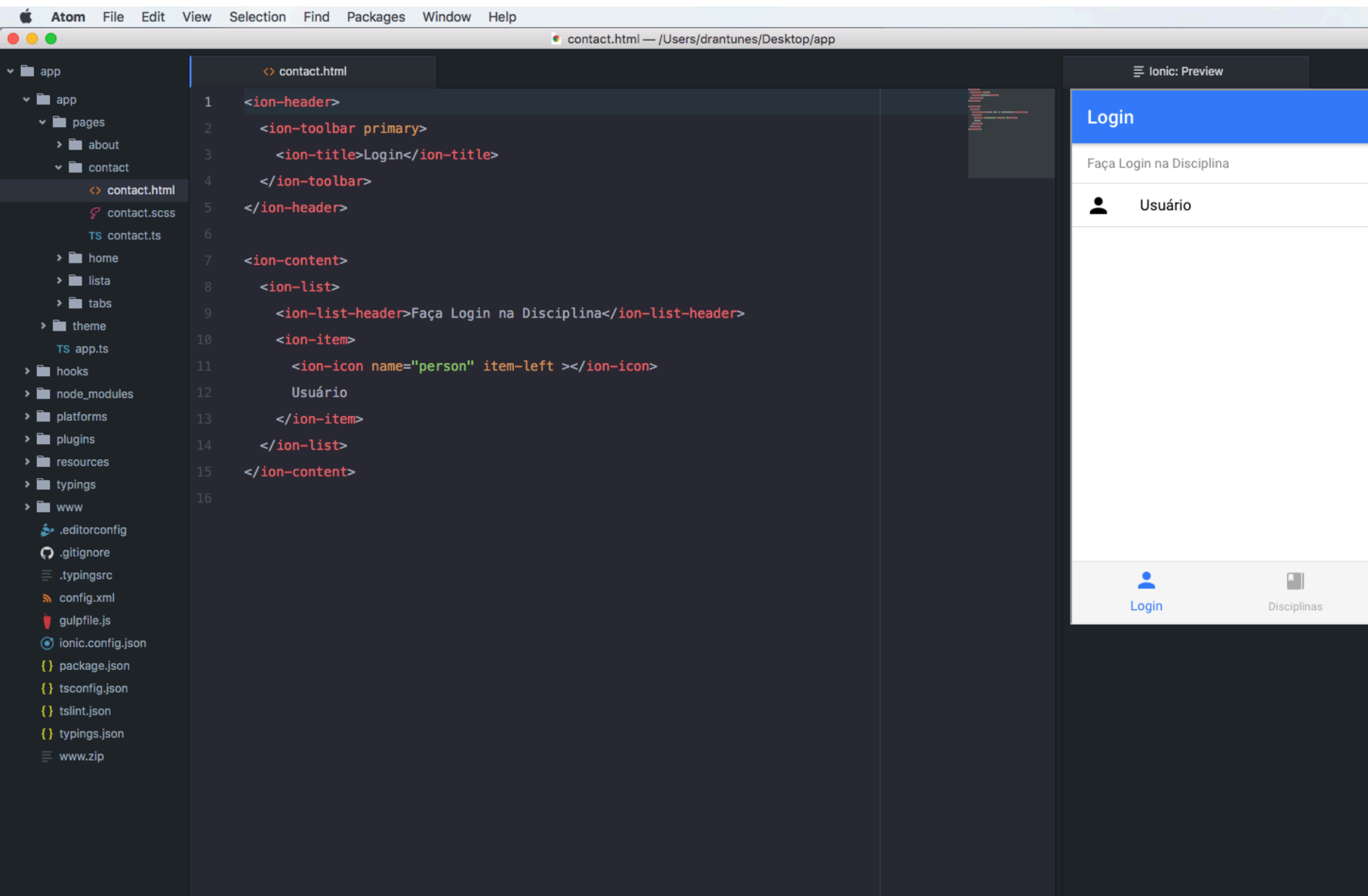
E carregar o ionic preview na direita



Ctrl + Alt + P: Ionic: Preview



Então o Ionic Preview será carregado



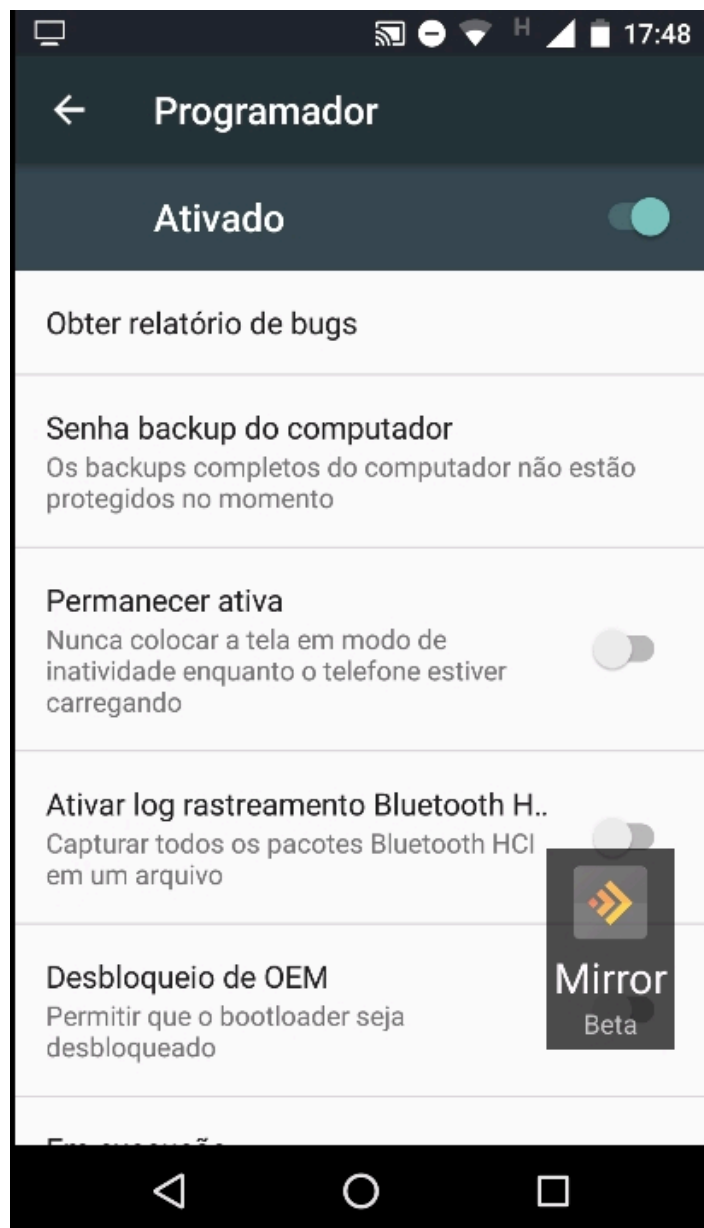
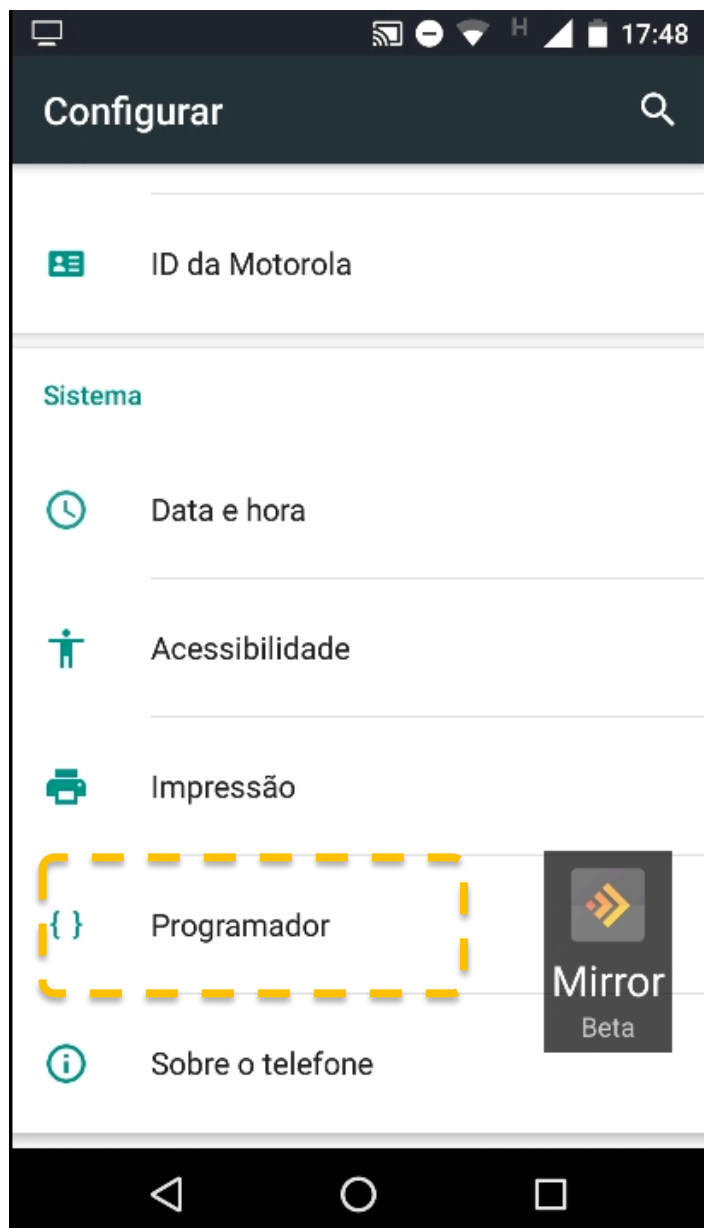
Como testar no
dispositivo físico

Habilitar Modo Programador

*Primeiro, você deve ir em seu smartphone Android, entrar no **Menu > Configurar > Sobre o Telefone** e clicar **5 vezes** sobre o item **Número da Versão**.*

Habilitar Modo Programador

*Esta ação irá liberar o modo desenvolvedor do seu Android e você poderá acessar a opção **Menu > Configurar > Programador***



Ativar Depuração USB

*Em seguida, dentro da opção Programador, você deve ativar a opção **Depuração via USB**.*



Programador

Ativado



Depuração

Depuração USB

Modo de depuração quando o USB estiver conectado



Revogar autorizações de depuração USB

Atalho para relatório de bugs

Mostrar um botão para gerar relatórios de bugs no menu do botão liga/desliga



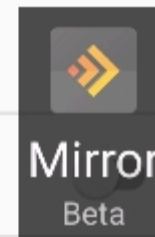
Selecionar app de local fictício

Nenhum app de local fictício definido

Visualizar inspeção de atributo

Selecionar aplicativo de depuração

Nenhum conjunto de aplicativos de depuração



Conecte seu Android via USB

Em seguida, conecte seu dispositivo Android ao seu Computador via USB.

Execute o projeto Ionic

*Então, basta você executar no Terminal ou CMD (Windows) o comando **ionic run android** Neste momento, o ionic irá fazer a compilação para o APK e fará a cópia para seu dispositivo físico.*

Execute o projeto Ionic

A primeira execução é um pouco demorada e depende de download de alguns pacotes adicionais via Internet.

Execute o projeto Ionic

Ao final, o ionic CLI irá abrir o aplicativo de maneira nativa em seu dispositivo físico via USB.

Ionic View

Ionic View

Com o Ionic View, você pode testar os aplicativos em seu dispositivo Android ou qualquer outro sem a necessidade de conectar o dispositivo via USB.

Ionic View

*Primeiro, você deve baixar o aplicativo **Ionic View** na Play Store ou App Store.*

<https://play.google.com/store/apps/details?id=com.ionic.viewapp>

Ionic View

Após instalar seu aplicativo você deve fazer o login com suas informações de cadastro no Ionic. Você pode fazer o cadastro pelo aplicativo ou no Ionic Cloud (<https://apps.ionic.io/signup>)

Ionic View

Para sincronizar seu aplicativo com o Ionic Cloud, no Terminal ou CMD (windows), dentro da pasta do projeto, você deve executar o comando:

ionic upload

Ionic View

Então, após sincronizar, basta você abrir em seu aplicativo Ionic View no celular.

Ionic View

Este recurso será utilizado para os alunos compartilharem seus trabalhos com o professor, para eventuais testes e acompanhamento dos exercícios.

Ionic View

*Devido aos problemas de proxy na UTFPR, inicialmente, o comando **ionic upload** deve ser realizado em casa.*

Estrutura de Pastas

Pastas

app: contém os arquivos de páginas, abas, menus, etc. que você está desenvolvendo. Dentro desta pasta, há o arquivo *app.ts* que consiste no arquivo de inicialização do seu aplicativo (como se fosse o main).

Pastas

app.ts este arquivo irá mudar se houver a necessidade de alterar sua página inicial ou adicionar alguma configuração opcional no aplicativo.

- ▼ app
 - ▼ pages
 - > about
 - > contact
 - > home
 - > lista
 - > tabs
 - > theme
- TS app.ts
- > hooks
- > node_modules
- > platforms
- > plugins
- > resources
- > typings
- > www
 - .editorconfig
 - .gitignore
 - .typingsrc
 - config.xml
 - gulpfile.js
 - ionic.config.json
 - { } package.json
 - { } tsconfig.json
 - { } tslint.json
 - { } typings.json
 - www.zip

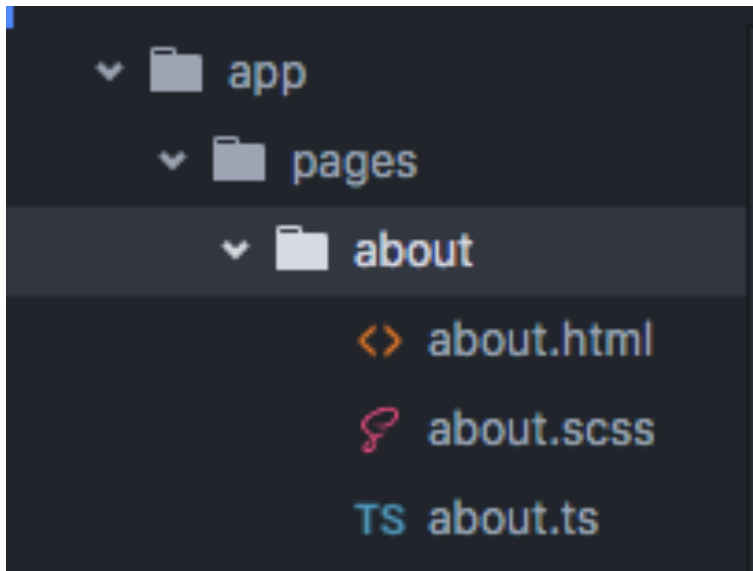
```
1  import {Component} from '@angular/core';
2  import {Platform, ionicBootstrap} from 'ionic-angular';
3  import {StatusBar} from 'ionic-native';
4  import {TabsPage} from '../pages/tabs/tabs';
5
6
7  @Component({
8    template: '<ion-nav [root]="rootPage"></ion-nav>'
9  })
10 export class MyApp {
11
12   private rootPage: any;
13
14   constructor(private platform: Platform) {
15     this.rootPage = TabsPage;
16
17     platform.ready().then(() => {
18       // Okay, so the platform is ready and our plugins are available.
19       // Here you can do any higher level native things you might need.
20       StatusBar.styleDefault();
21     });
22   }
23 }
24
25 ionicBootstrap(MyApp);
26
```

Pasta Pages

pages esta pasta lista todos os componentes de sua aplicação, bem como suas páginas (pages). Por exemplo, no modelo inicial são geradas as páginas de Contato, Home e About.

Pasta Pages

*Cada componente (página) possui três arquivos principais: **nome_da_pagina.scss**, **.html** e **.ts**.*



HTML: Componentes Visuais;

TS: Código JavaScript;

SCSS: Código CSS Personalizado.

Pasta Hooks

Comandos especiais inseridos pelo ionic ou terceiros, para auxiliar na compilação. Não é necessário mexer.

Pasta node_modules

Módulos NPM instalados localmente para serem usados pelo aplicativo híbrido, por exemplo, o Angular.

Pasta platforms

*Contém os arquivos específicos de cada plataforma que você adicionou ao aplicativo, usando o comando **ionic platform add android***

Pasta platforms

*Dentro desta pasta também será gerado o APK quando você executar um comando de **build** no ionic.*

Pasta plugins

Contém os plugins (instalados via Apache Cordova) para acessar os recursos nativos.

Pasta resources

Contém as imagens de splash screen e ícones para seu aplicativo, para cada resolução de tela.

Pasta typings

Módulo especial usado pelo ionic para compatibilizar bibliotecas ES5 para a Sintaxe ES6 ou TypeScript.

Pasta www

Contém os arquivos gerados para o teste no navegador.

Componentes Visuais

Button

Na documentação existem formas diferentes de usar um botão, como alterar cor, tamanho, estilo, borda, etc.

<http://ionicframework.com/docs/v2/components/#buttons>

Button

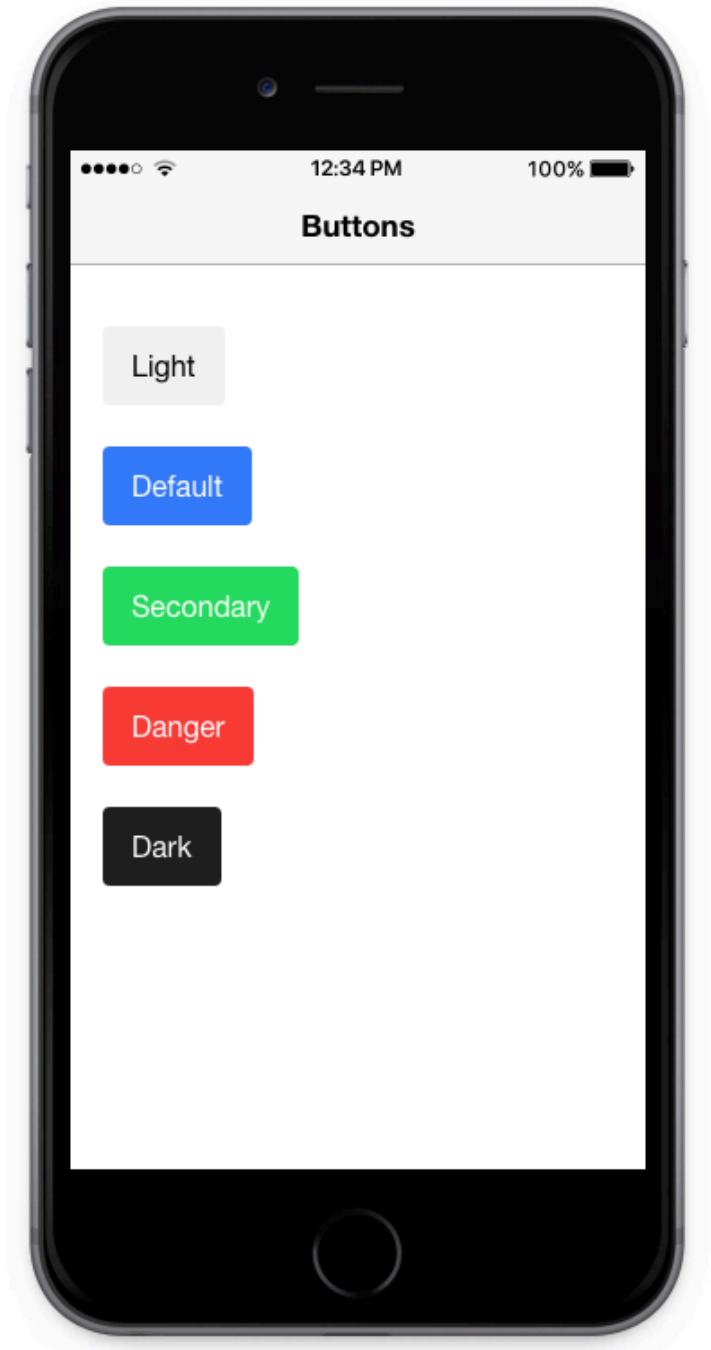
```
<button light>Light</button>
```

```
<button>Default</button>
```

```
<button secondary>Secondary</button>
```

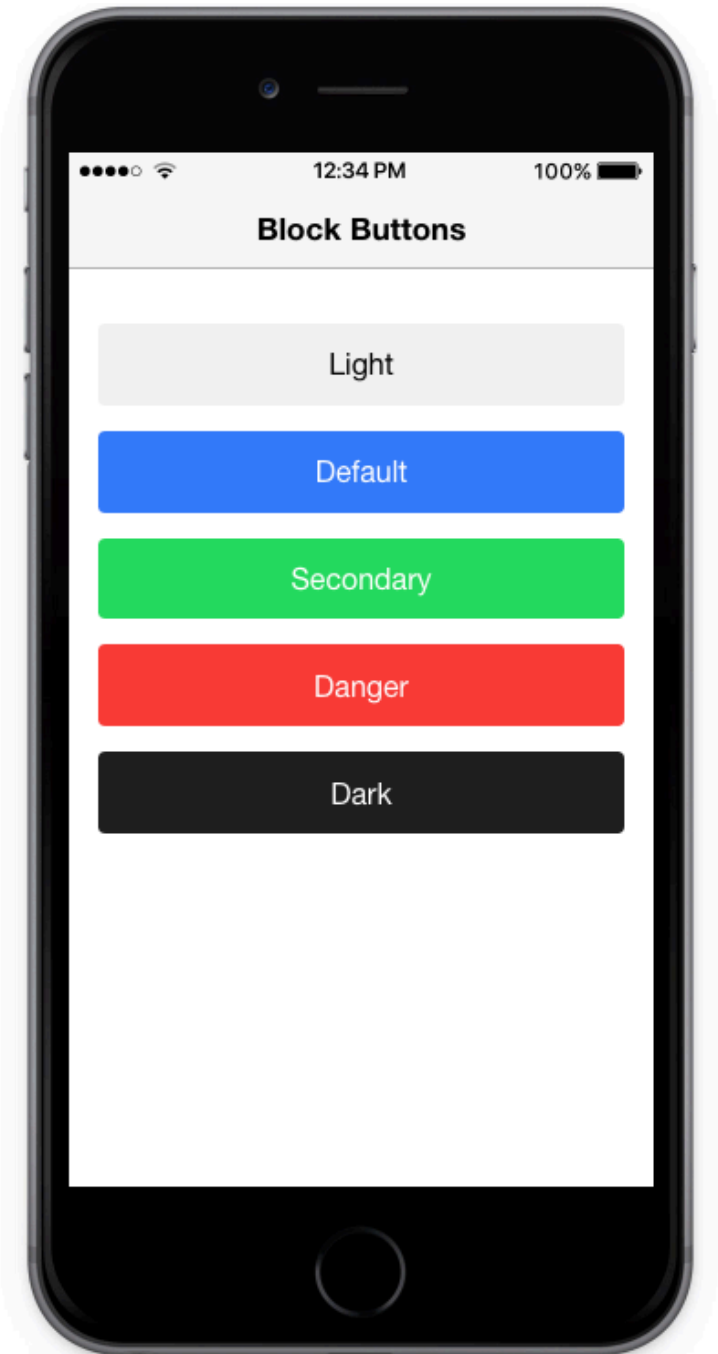
```
<button danger>Danger</button>
```

```
<button dark>Dark</button>
```



Button

```
<button block>Block Button</button>
```

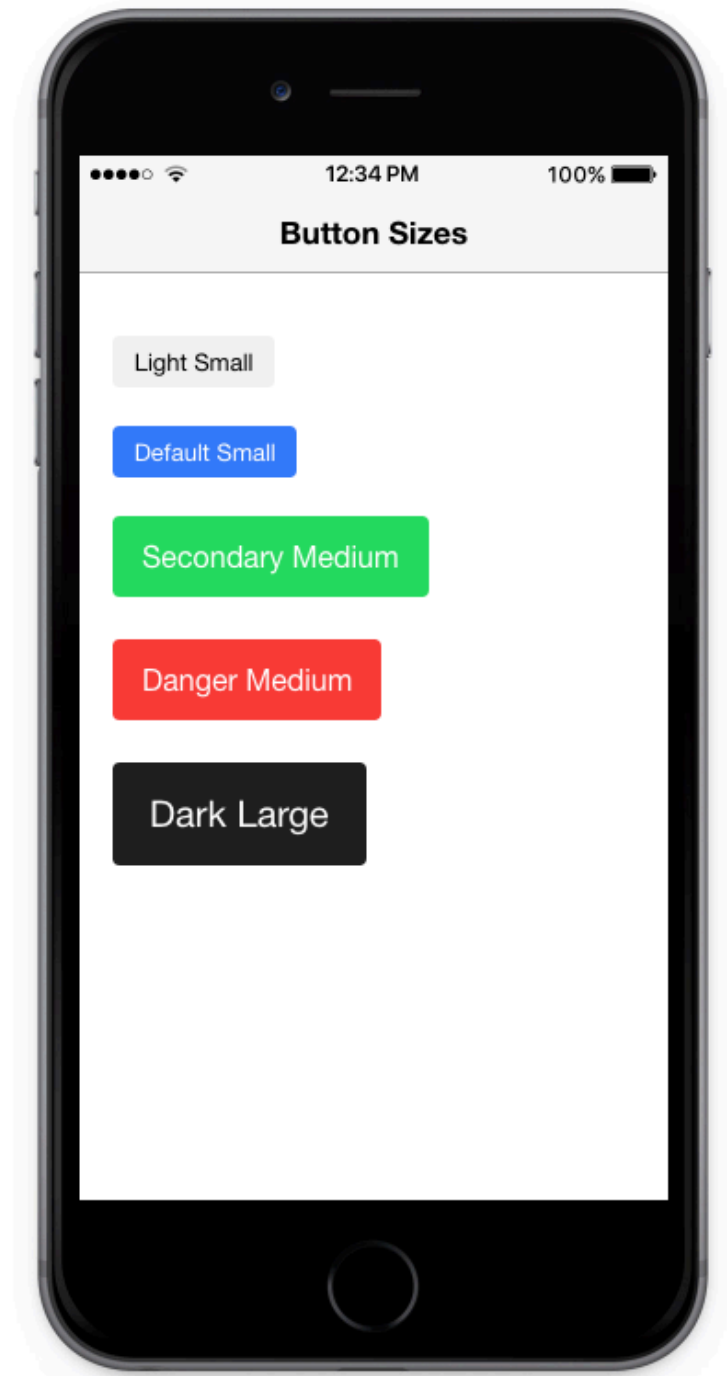


Button

```
<button small>Small</button>
```

```
<button>Default</button>
```

```
<button large>Large</button>
```



Input e Label

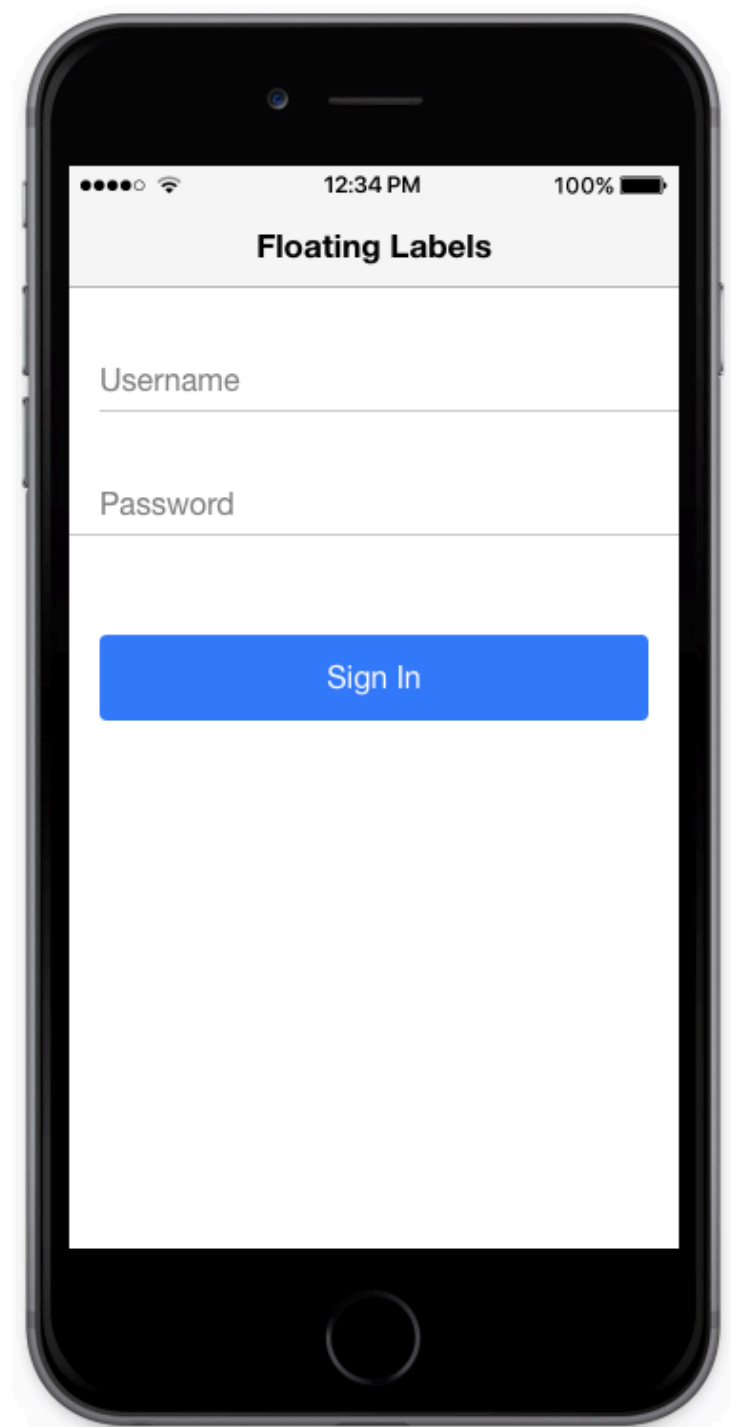
Existem diferentes combinações de campos de texto e labels: fixo, float, inline, placeholder, entre outros. A opção floating labels é uma ótima opção.

<http://ionicframework.com/docs/v2/components/#inputs>

Input e Label

```
<ion-item>  
  <ion-label floating>Username</ion-label>  
  <ion-input type="text"></ion-input>  
</ion-item>
```

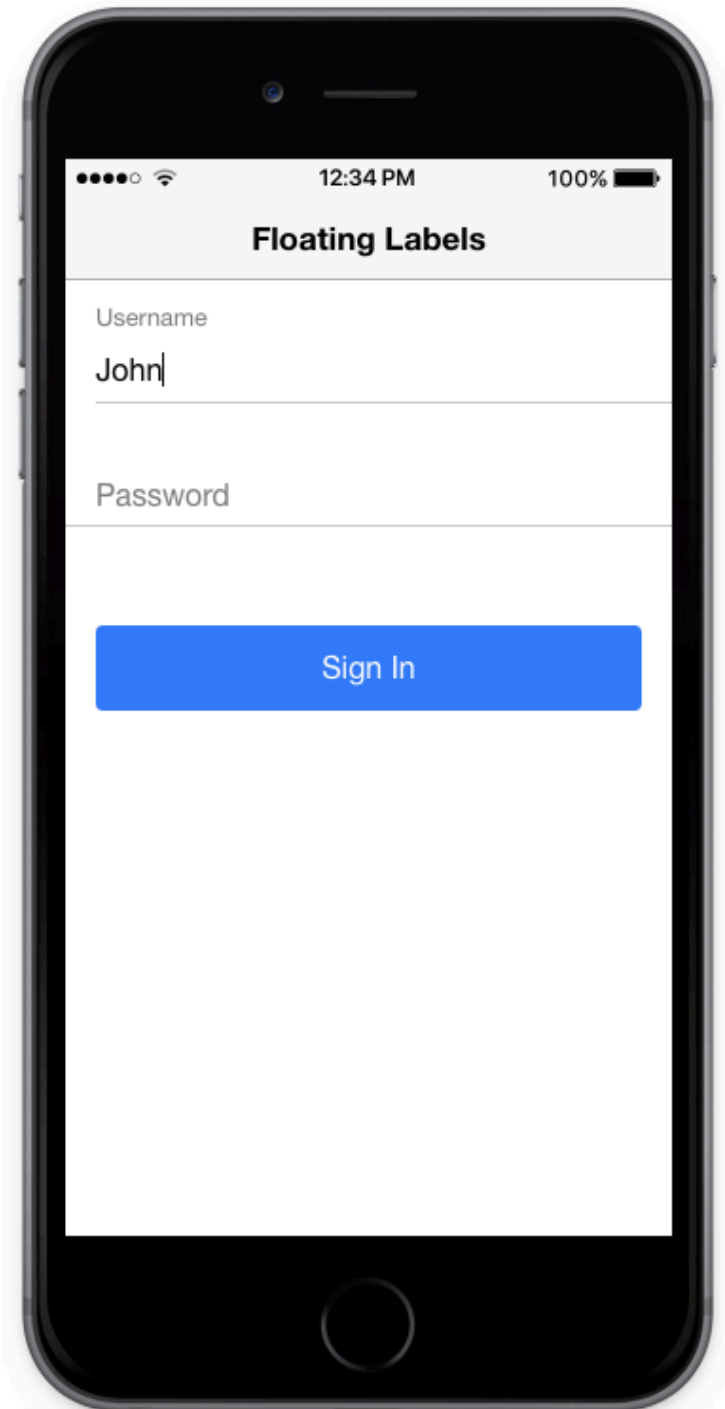
```
<ion-item>  
  <ion-label floating>Password</ion-label>  
  <ion-input type="password"></ion-input>  
</ion-item>
```



Input e Label

```
<ion-item>  
  <ion-label floating>Username</ion-label>  
  <ion-input type="text"></ion-input>  
</ion-item>
```

```
<ion-item>  
  <ion-label floating>Password</ion-label>  
  <ion-input type="password"></ion-input>  
</ion-item>
```



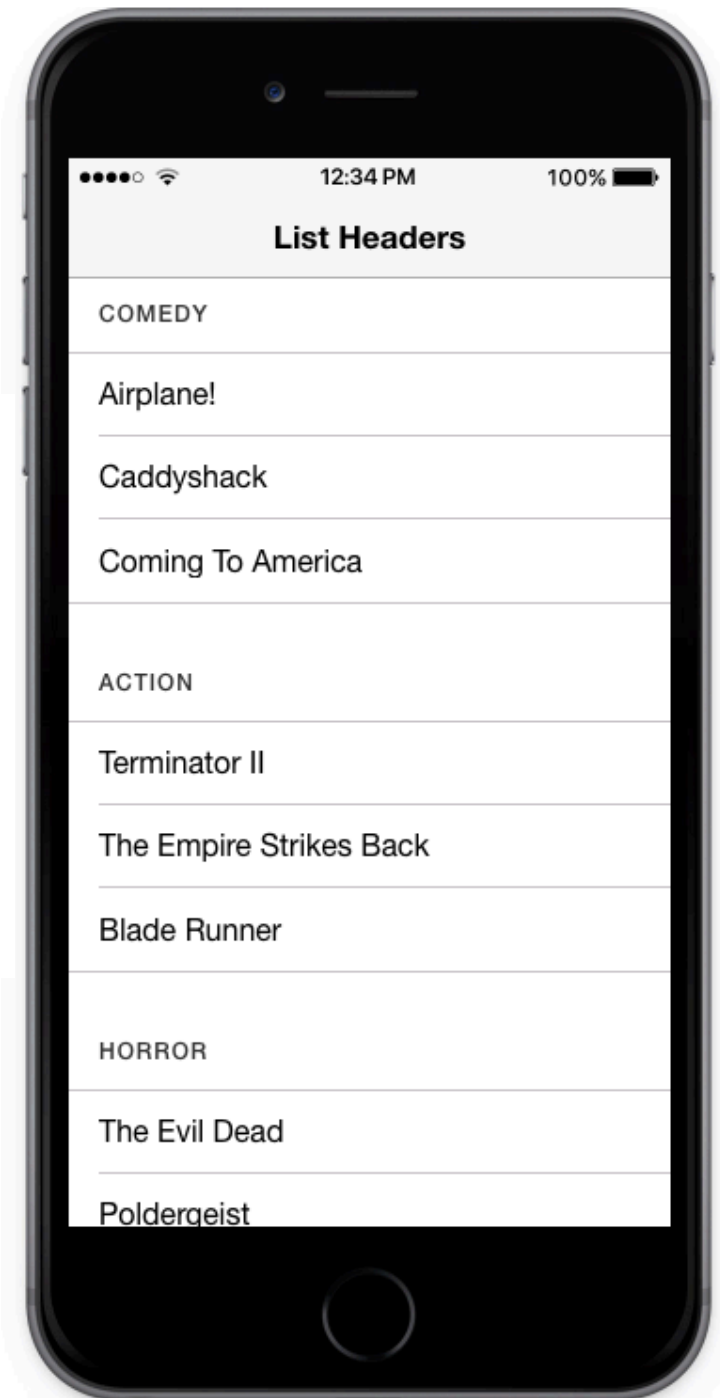
Lists

Um componente necessário para listar informações na tela de um aplicativo. Você pode combinar listas com textos, fotos, ícones, sliding listings, entre outros.

<http://ionicframework.com/docs/v2/components/#lists>

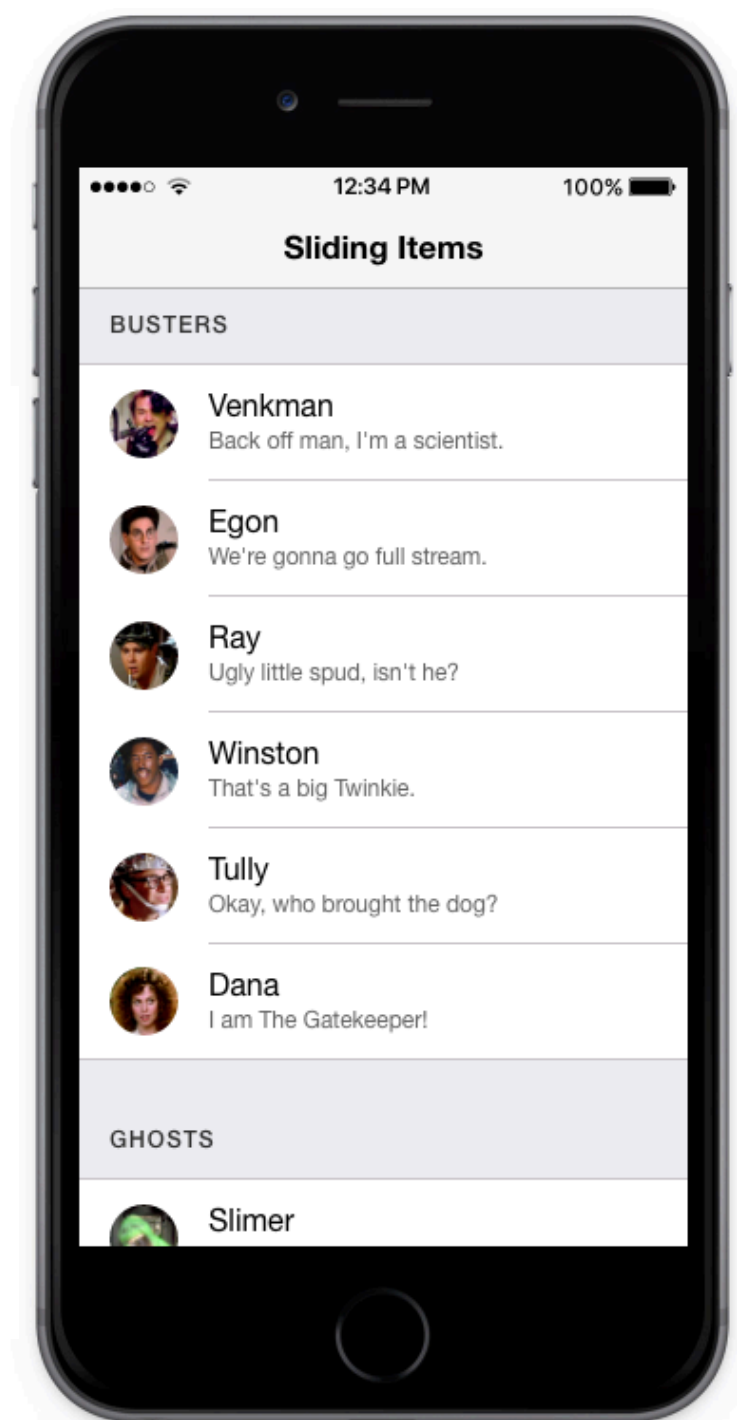
Head List

```
<ion-list>
  <ion-list-header>
    Action
  </ion-list-header>
  <ion-item>Terminator II</ion-item>
  <ion-item>The Empire Strikes Back</ion-item>
  <ion-item>Blade Runner</ion-item>
</ion-list>
```



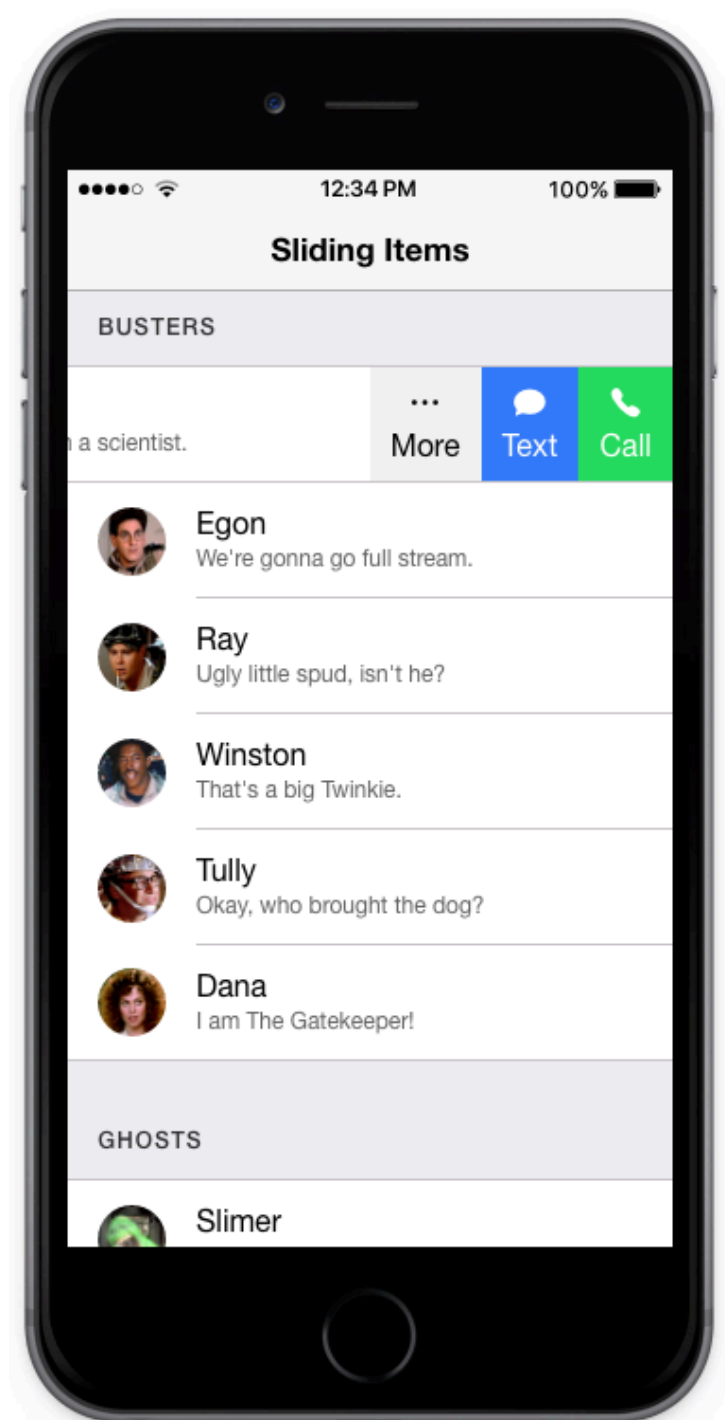
Slide List

```
<ion-list>
  <ion-item-sliding>
    <ion-item>
      <ion-avatar item-left>
        
      </ion-avatar>
      <h2>Slimer</h2>
    </ion-item>
    <ion-item-options side="left">
      <button primary>
        <ion-icon name="text"></ion-icon>
        Text
      </button>
    </ion-item-options>
  </ion-item-sliding>
</ion-list>
```



Slide List

```
<ion-list>
  <ion-item-sliding>
    <ion-item>
      <ion-avatar item-left>
        
      </ion-avatar>
      <h2>Slimer</h2>
    </ion-item>
    <ion-item-options side="left">
      <button primary>
        <ion-icon name="text"></ion-icon>
        Text
      </button>
    </ion-item-options>
  </ion-item-sliding>
</ion-list>
```



Checkbox

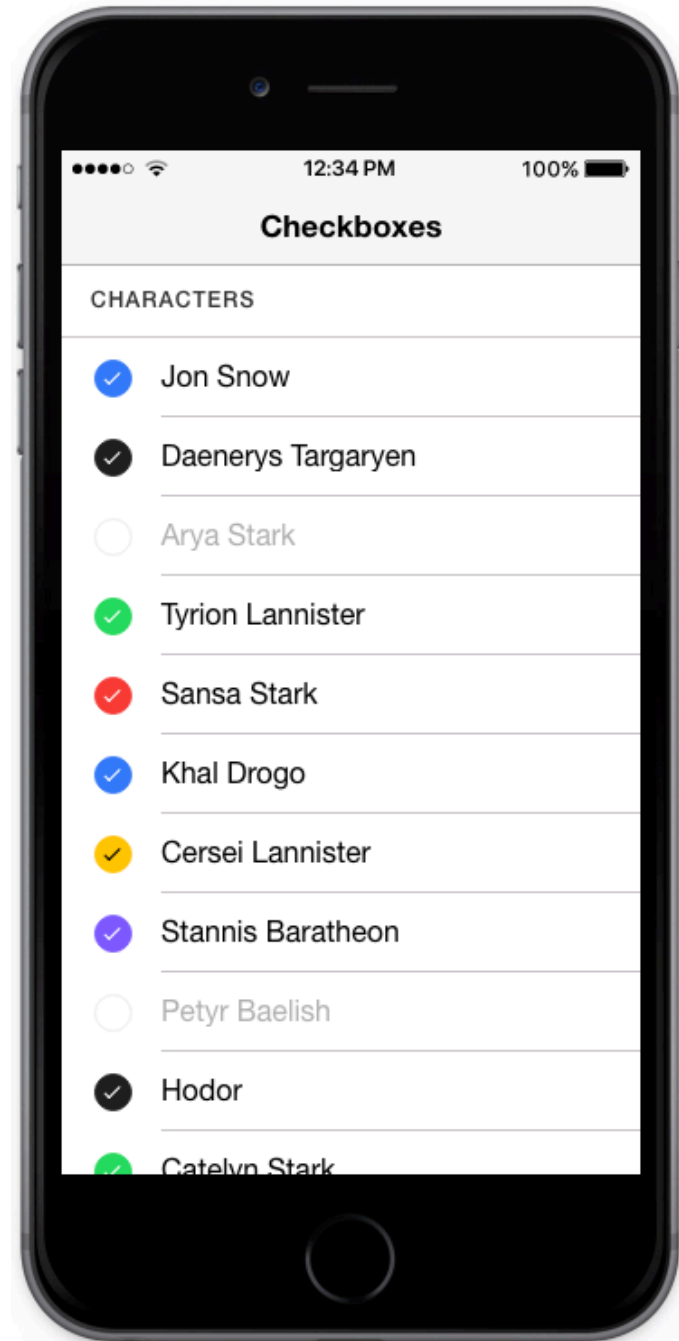
Um componente necessário marcar itens. Por exemplo, para uma lista de tarefas a fazer ou tarefas completas.

<http://ionicframework.com/docs/v2/components/#checkbox>

Checkbox

```
<ion-item>  
  <ion-label>Daenerys Targaryen</ion-label>  
  <ion-checkbox dark checked="true"></ion-checkbox>  
</ion-item>
```

```
<ion-item>  
  <ion-label>Arya Stark</ion-label>  
  <ion-checkbox disabled="true"></ion-checkbox>  
</ion-item>
```



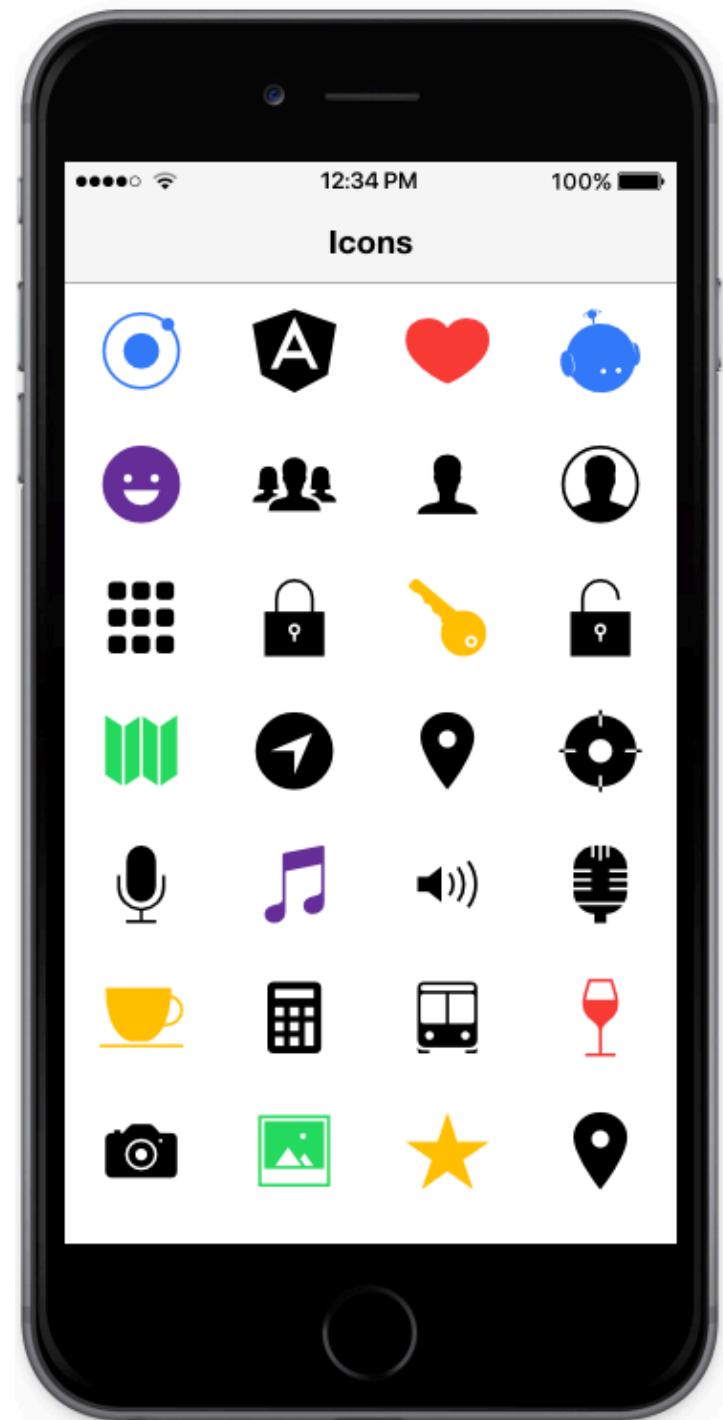
Ícones

Você pode utilizar um componente específico do ionic para inserir ícones em seu aplicativo. Você pode colorir estes ícones, usar em listas, botões, etc.

<http://ionicframework.com/docs/v2/components/#icons>

Icons

```
<ion-icon name="heart"></ion-icon>
```

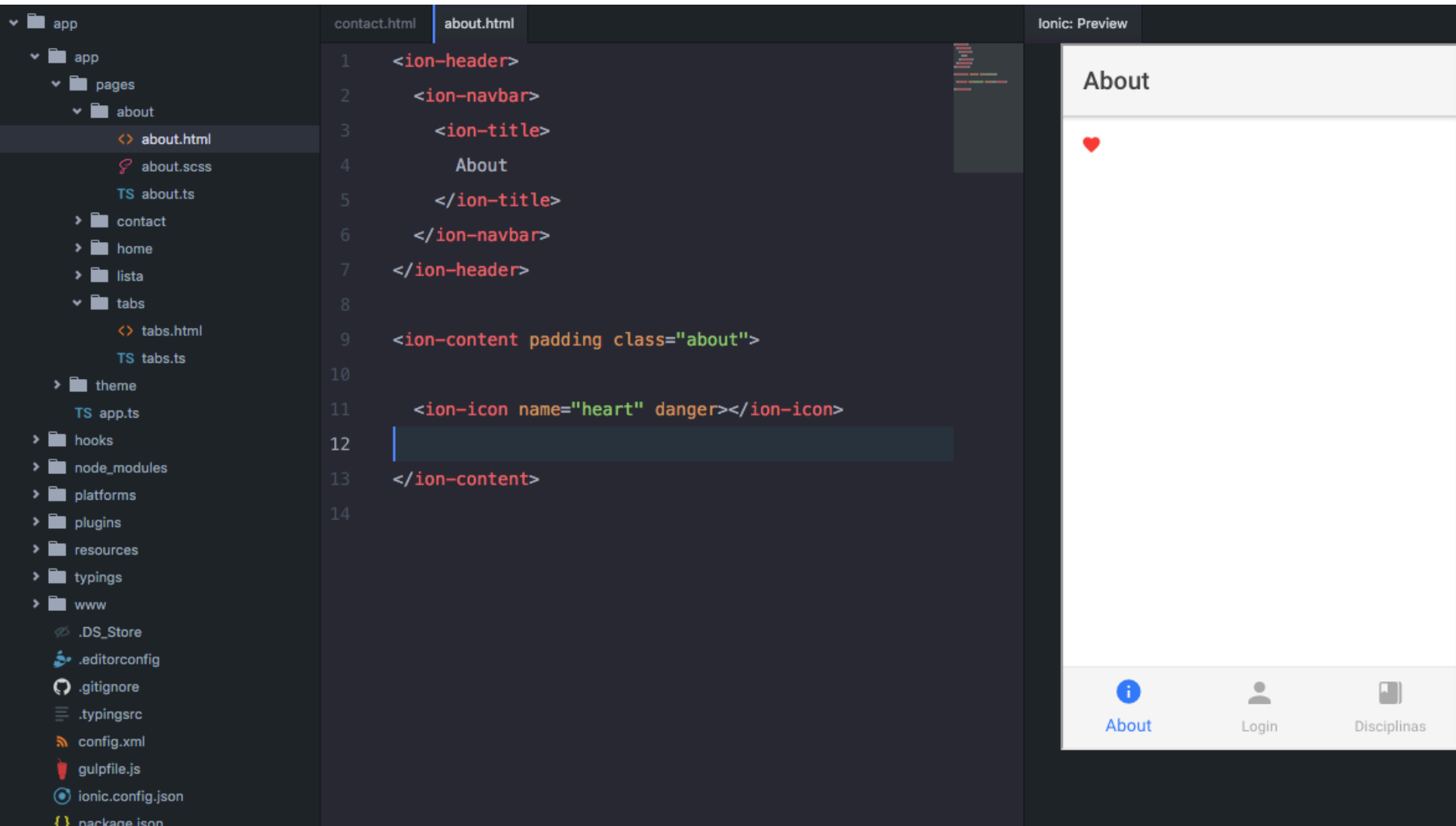


Onde inserir os componentes?

Testando os componentes

Você irá adicionar os componentes visuais do ionic dentro das páginas HTML do seu aplicativo.

Ex: Ícone na Página About



The image shows a code editor with a file explorer on the left, a code editor in the center, and a preview window on the right.

File Explorer (Left):

- app
 - pages
 - about
 - about.html (selected)
 - about.scss
 - about.ts
 - contact
 - home
 - lista
 - tabs
 - tabs.html
 - tabs.ts
 - theme
 - app.ts
 - hooks
 - node_modules
 - platforms
 - plugins
 - resources
 - typings
 - www
 - .DS_Store
 - .editorconfig
 - .gitignore
 - .typingsrc
 - config.xml
 - gulpfile.js
 - ionic.config.json
 - package.json

Code Editor (Center):

```
1 <ion-header>
2   <ion-navbar>
3     <ion-title>
4       About
5     </ion-title>
6   </ion-navbar>
7 </ion-header>
8
9 <ion-content padding class="about">
10
11   <ion-icon name="heart" danger></ion-icon>
12
13 </ion-content>
14
```

Preview Window (Right):

The preview window shows the rendered page titled "About". It features a red heart icon in the center of the page. The bottom of the preview shows a navigation bar with three items: "About" (selected), "Login", and "Disciplinas".

Aqui é o cabeçalho da página. Você pode usar o *ion-navbar* ou o *ion-toolbar* (ver documentação)

The image shows a code editor with a file explorer on the left and a preview window on the right. The file explorer shows a project structure with folders like 'app', 'pages', 'contact', 'home', 'lista', 'tabs', and 'theme'. The 'about.html' file is selected in the 'pages/about' folder. The code editor shows the following HTML code for 'about.html':

```
1 <ion-header>
2   <ion-navbar>
3     <ion-title>
4       About
5     </ion-title>
6   </ion-navbar>
7 </ion-header>
8
9 <ion-content padding class="about">
10
11   <ion-icon name="heart" danger></ion-icon>
12
13 </ion-content>
14
```

The code is highlighted with a dashed yellow box. The preview window on the right shows the rendered page titled 'About'. It features a red heart icon in the content area and a bottom bar with three buttons: 'About' (with an information icon), 'Login' (with a user icon), and 'Disciplinas' (with a book icon).

Aqui é o conteúdo da página. Todos os componentes devem ser inseridos dentro do *ion-content*.

The image shows a code editor interface for an Ionic application. The left sidebar displays the project structure, with the 'about.html' file selected under the 'pages/about' directory. The main editor area shows the HTML code for 'about.html', which is structured as follows:

```
1 <ion-header>
2   <ion-navbar>
3     <ion-title>
4       About
5     </ion-title>
6   </ion-navbar>
7 </ion-header>
8
9 <ion-content padding class="about">
10
11   <ion-icon name="heart" danger></ion-icon>
12
13 </ion-content>
14
```

A dashed yellow box highlights the `<ion-content>` block, indicating where page content should be placed. The right sidebar shows a preview of the application, titled 'About', displaying a red heart icon. The bottom of the preview shows a tab bar with three tabs: 'About' (active), 'Login', and 'Disciplinas'.

Trabalho 1: Moodle

Lista 1 (Parte 1): Moodle

Extras

1. *Testar os componentes visuais do Ionic em um aplicativo de exemplo;*
2. *Tentar alterar cores, posicionamento, estilos de acordo com os exemplos apresentados na documentação;*