

Roteiro: Criando o primeiro APP

1 Instale o ambiente conforme o documento “Configuração do ambiente”

2 Abra o ícone/atalho “Node command prompt” e digite o comando “ionic start app1”

Este comando criará a estrutura de pastas de um novo aplicativo “app1”

Como Framework escolha Angular

Escolha Blank como modelo

Depois escolha StandAloneComponents

```
luizmaia@macbook Mobile % ionic start app1

Pick a framework! 🤖

Please select the JavaScript framework to use for your new app. To bypass this
prompt next time, supply a value for the --type option.

? Framework: Angular

Let's pick the perfect starter template! 🍌

Starter templates are ready-to-go Ionic apps that come packed with everything
you need to build your app. To bypass this prompt next time, supply template,
the second argument to ionic start.

? Starter template: blank
? Would you like to build your app with NgModules or Standalone Components?
Standalone components are a new way to build with Angular that simplifies the way you build your app.
To learn more, visit the Angular docs:
https://angular.io/guide/standalone-components

Standalone
✓ Preparing directory ./app in 1.30ms
✓ Downloading and extracting blank starter in 589.46ms
> ionic integrations enable capacitor --quiet -- app io.ionic.starter
> npm i --save -E @capacitor/core@latest
```

Este comando de criação é um pouco demorado, pois ele faz o download de pacotes complementares. Favor aguardar a finalização.

```
? Create free Ionic account? No

Your Ionic app is ready! Follow these next steps:

- Go to your new project: cd ./app
- Run ionic serve within the app directory to see your app in the browser
- Run ionic capacitor add to add a native iOS or Android project using Capacitor
- Generate your app icon and splash screens using cordova-res --skip-config --copy
- Explore the Ionic docs for components, tutorials, and more:
https://ion.link/docs
- Building an enterprise app? Ionic has Enterprise Support and Features:
https://ion.link/enterprise-edition
luizmaia@macbook Mobile %
```

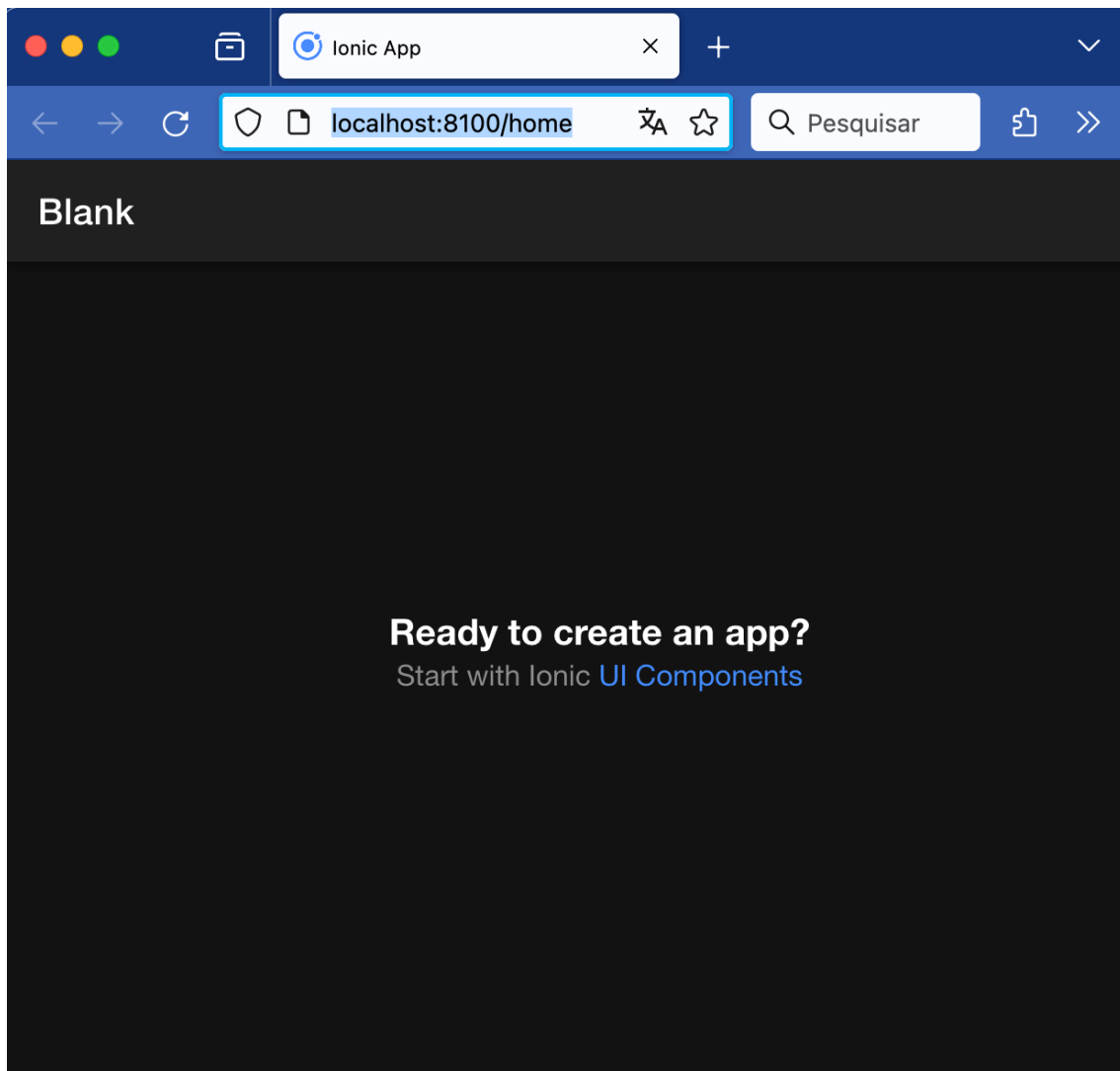
Mude para o diretório do projeto com o comando “cd app”

Visualize o aplicativo no navegador com o comando “ionic serve”

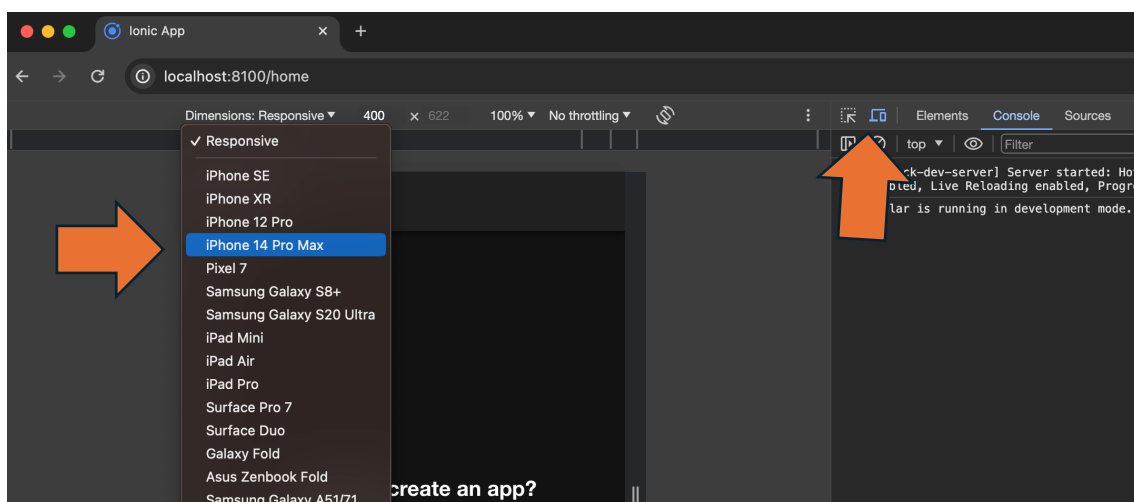
```
luizmaia@macbook app % ionic serve
> ng run app:serve --host=localhost --port=8100
[ng] - Generating browser application bundles (phase: setup)...
```

Após gerar os arquivos necessários para a visualização, o navegador deverá abrir no site: <http://localhost:8100/home>

O Ionic abre a porta 8100 da máquina.



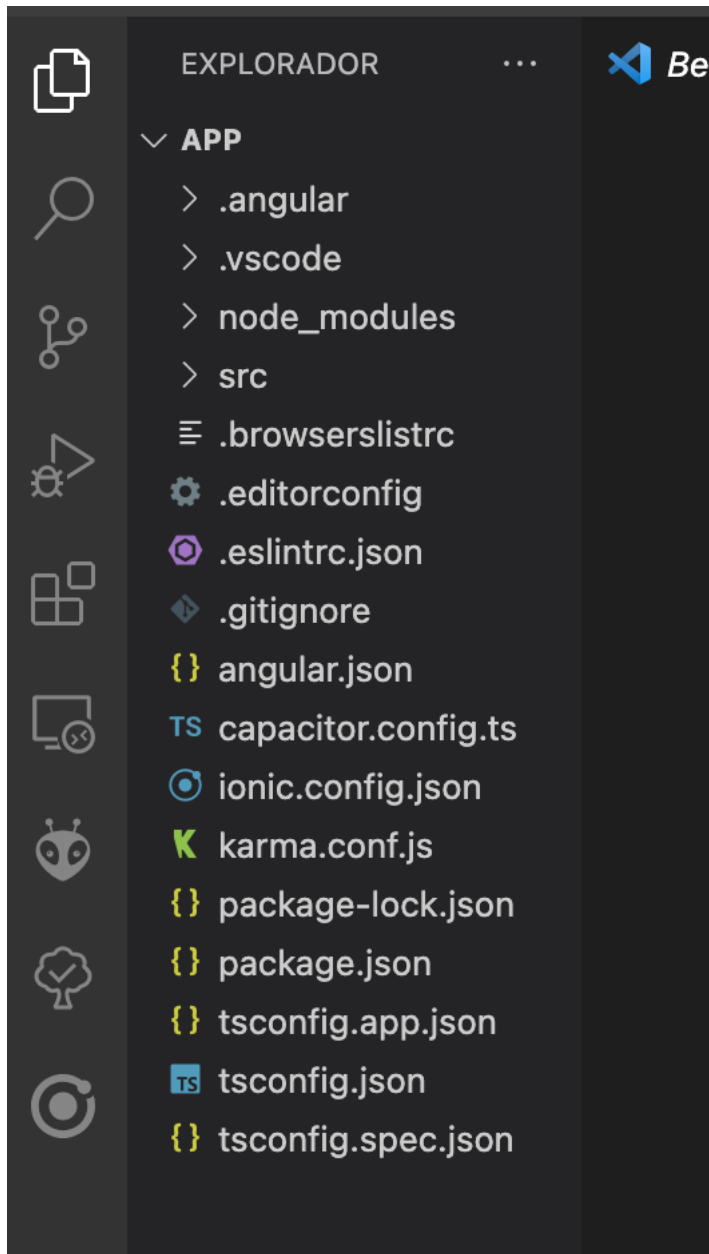
Vamos visualizar o aplicativo no formato de aplicativo de celular
Entre no modo de inspeção do navegador (Ctrl+Shift+I)



Clique no segundo ícone escolha um celular para visualização (Iphone / Samsung Galaxy).

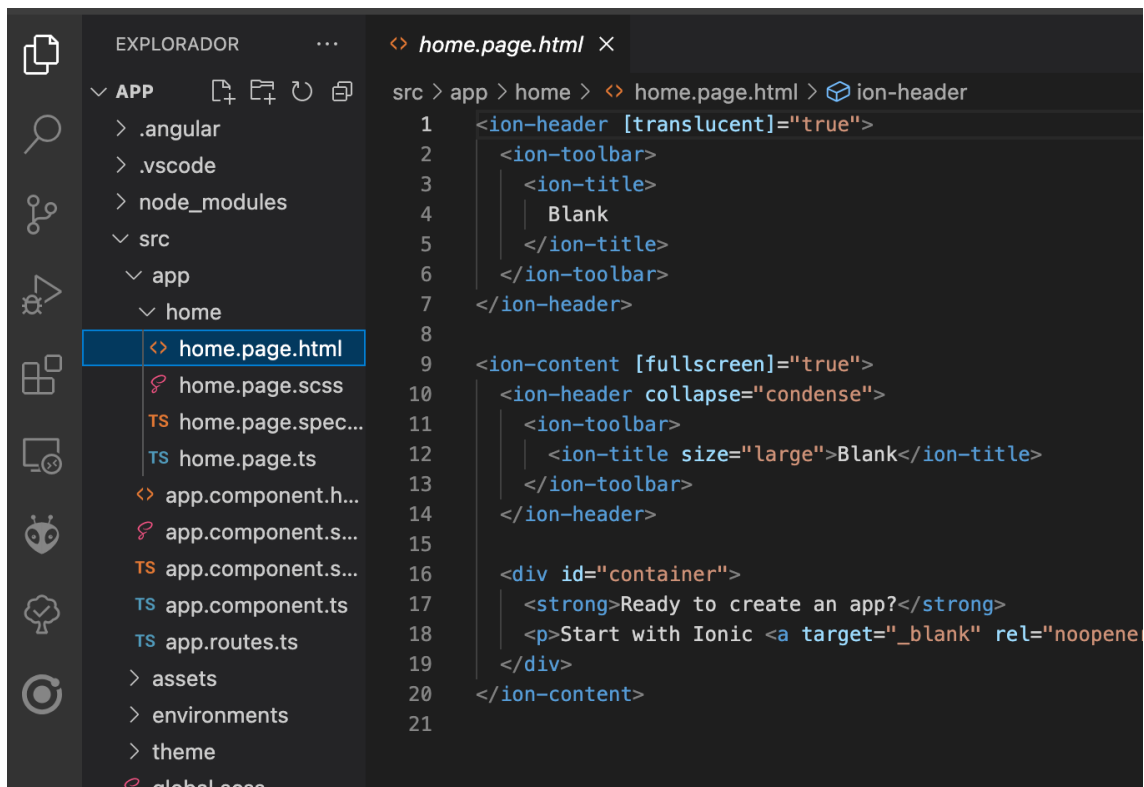
Abra o Visual Studio Code, escolha a opção “Abrir pasta...” e escolha a pasta do aplicativo criado.

A estrutura de diretórios será aberta.



Vamos modificar a tela inicial, para isto abra o arquivo

src/app/home.page.html



```
1 <ion-header [translucent]="true">
2   <ion-toolbar>
3     <ion-title>
4       Blank
5     </ion-title>
6   </ion-toolbar>
7 </ion-header>
8
9 <ion-content [fullscreen]="true">
10   <ion-header collapse="condense">
11     <ion-toolbar>
12       <ion-title size="large">Blank</ion-title>
13     </ion-toolbar>
14   </ion-header>
15
16   <div id="container">
17     <strong>Ready to create an app?</strong>
18     <p>Start with Ionic <a target="_blank" rel="noopener">
19   </div>
20 </ion-content>
21
```

Mude o Título do aplicativo de Blank para Primeiro Aplicativo

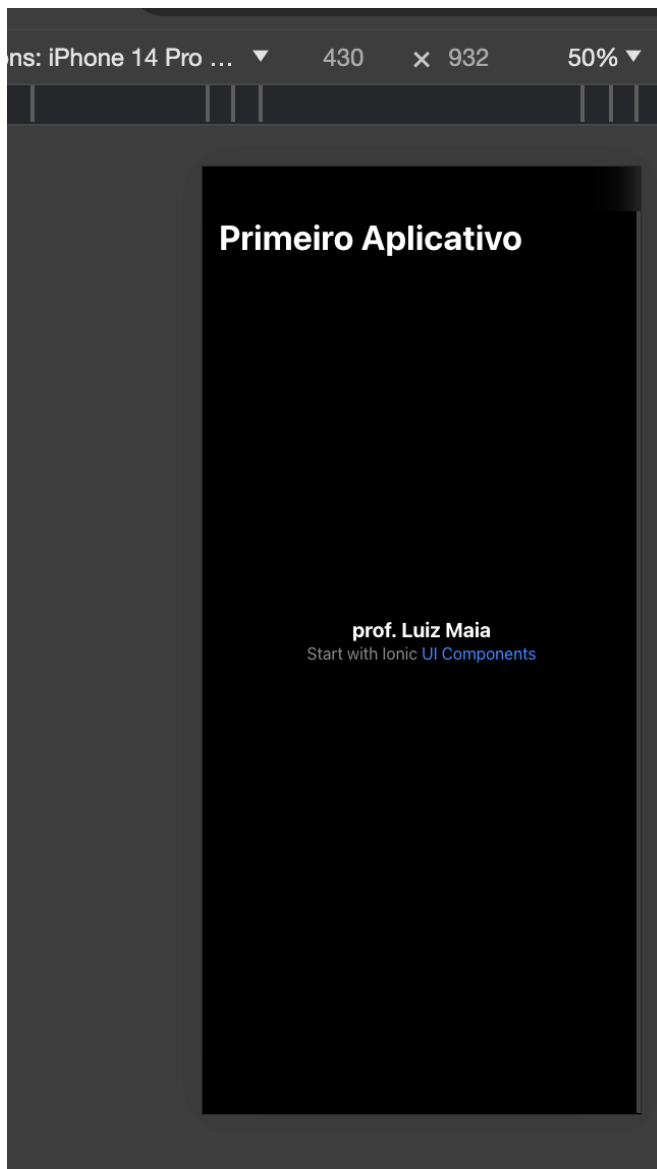
Mude a mensagem dentro da div container:

Ready to create an app?

Para o nome dos alunos que estão realizando este roteiro.

Salve o arquivo e retorne ao navegador.

Veja se as alterações foram realizadas:



Conforme visto em sala o Ionic possui diversos componentes que permitem compor as telas. Nesta parte, incluiremos um componente na tela inicial.

Clique no link do aplicativo “UI Components”

Inicialmente vamos adicionar um botão na tela.

Clique no Ion Button e verifique o código para inserção na página escolhendo “angular”. Copie o código e cole entre as divs `` e `<p>` no código da home page. Troque o nome do primeiro botão para Iniciar e o do segundo para Retornar. Observe que o segundo botão estará desabilitado inicialmente. Salve o código.

Verifique o navegador, você verá um erro sobre a ausência da importação de componentes.

Temos que importar o novo componente no arquivo [home.page.ts](#).

```

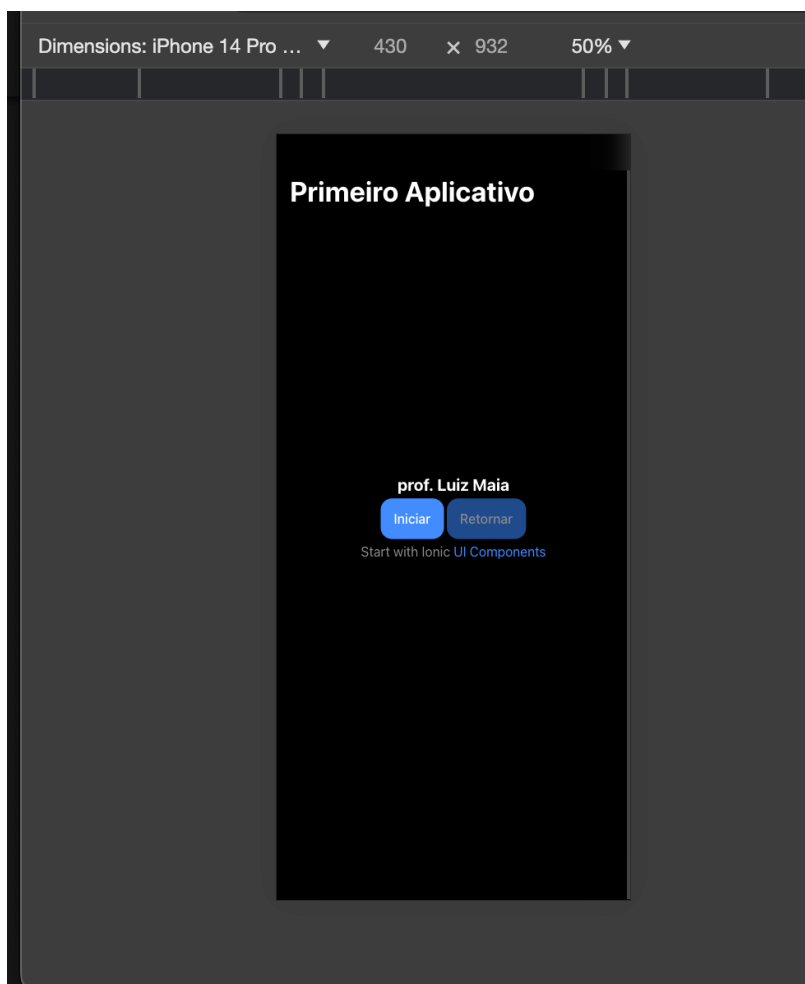
app > home > ts > home.page.ts > home.page
import { Component } from '@angular/core';
import { IonHeader, IonToolbar, IonTitle, IonContent, IonButton } from '@ionic/angular/

@Component({
  selector: 'app-home',
  templateUrl: 'home.page.html',
  styleUrls: ['home.page.scss'],
  standalone: true,
  imports: [IonHeader, IonToolbar, IonTitle, IonContent, IonButton],
})
class HomePage
export class HomePage {
  constructor() {}
}

```

Modifique o arquivo TypeScript da página principal conforme a imagem acima, inserindo a importação do componente botão. Salve o arquivo e retorne ao navegador.

Você terá finalizado esta parte do roteiro se encontrar a seguinte tela:



Seguindo e repetindo os passos acima, vamos inserir um componente de ícone em nosso aplicativo, logo abaixo dos botões.

Entre no site:

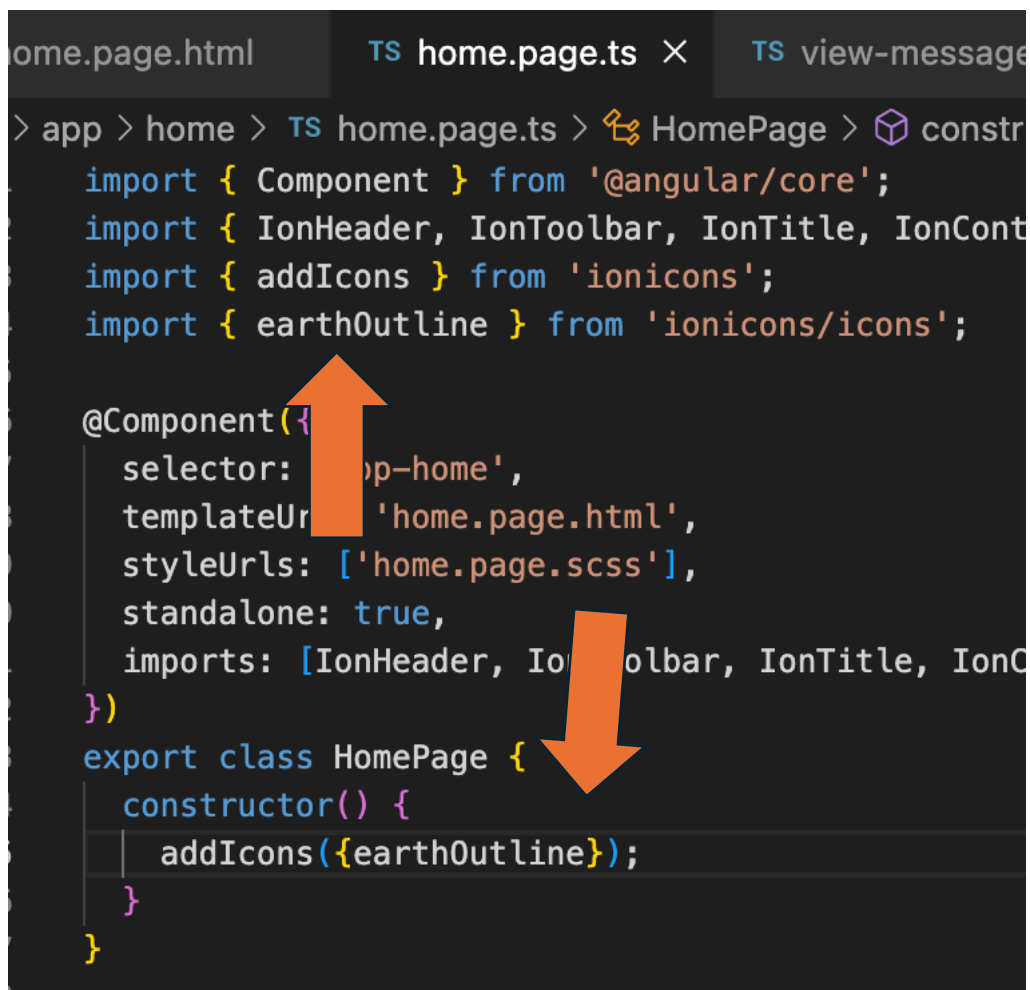
<https://ionicframework.com/docs/api/icon>

e clique para visualizar a lista de ícones. Escolha um ícone e visualize seu código:

```
<ion-icon name="NOME DO ÍCONE ESCOLHIDO" size="large"></ion-icon>
```

Insira o código do ícone escolhido junto ao código html da página principal, logo abaixo dos botões.

Também é necessário importar os ícones e adicioná-los junto ao constructor da página. Faça a adaptação do nome do ícone, conforme abaixo.

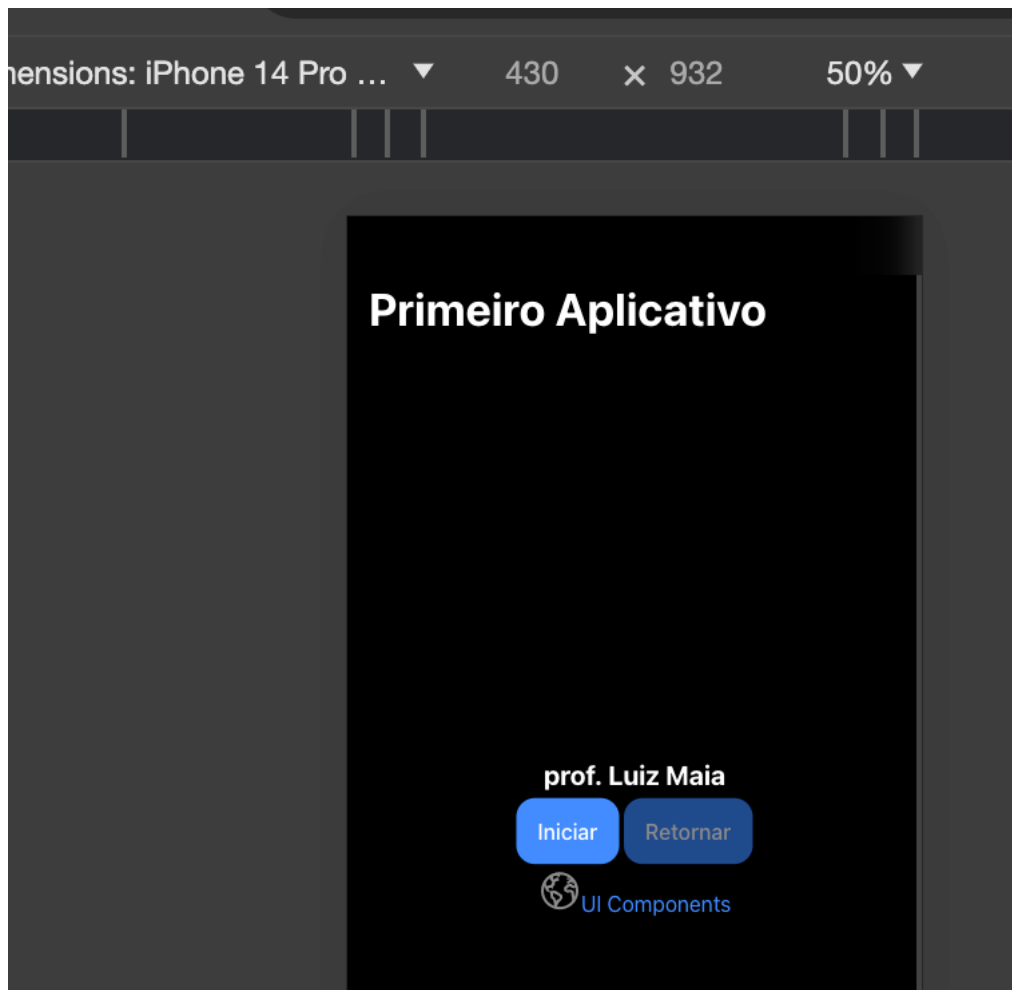


```
ome.page.html    TS home.page.ts X    TS view-message

> app > home > TS home.page.ts > 📁 HomePage > 📦 constr
import { Component } from '@angular/core';
import { IonHeader, IonToolbar, IonTitle, IonCont
import { addIcons } from 'ionicons';
import { earthOutline } from 'ionicons/icons';

@Component({
  selector: 'app-home',
  templateUrl: 'home.page.html',
  styleUrls: ['home.page.scss'],
  standalone: true,
  imports: [IonHeader, IonToolbar, IonTitle, IonC
})
export class HomePage {
  constructor() {
    addIcons({earthOutline});
  }
}
```

Você terá sucesso nesta parte do roteiro se a tela estive parecida com a abaixo:



Tire um print comprovando a finalização desta parte do roteiro.