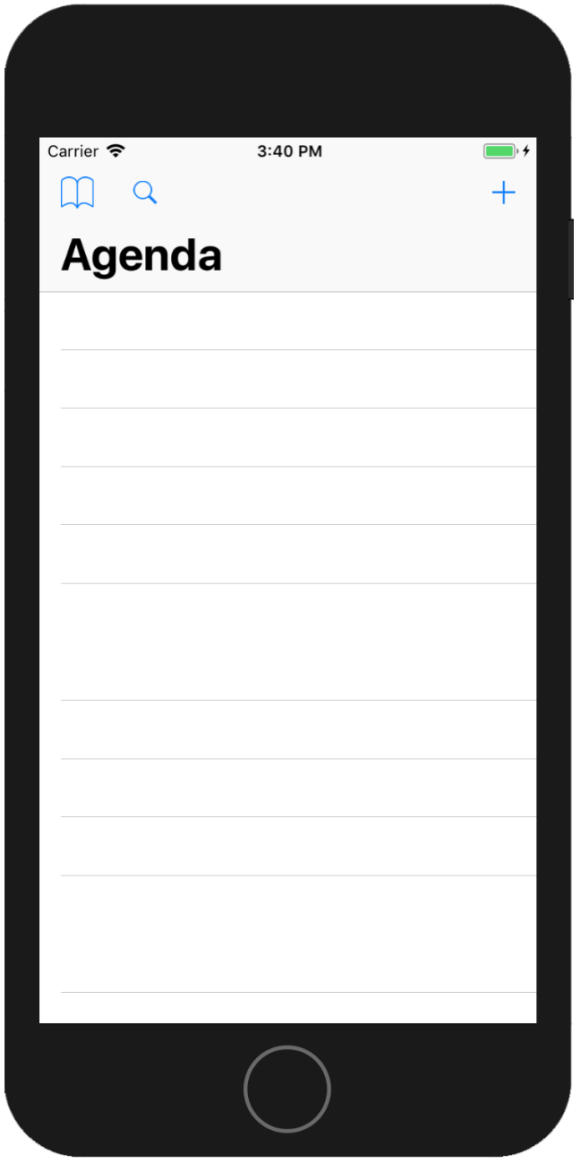
03

Acessando a câmera

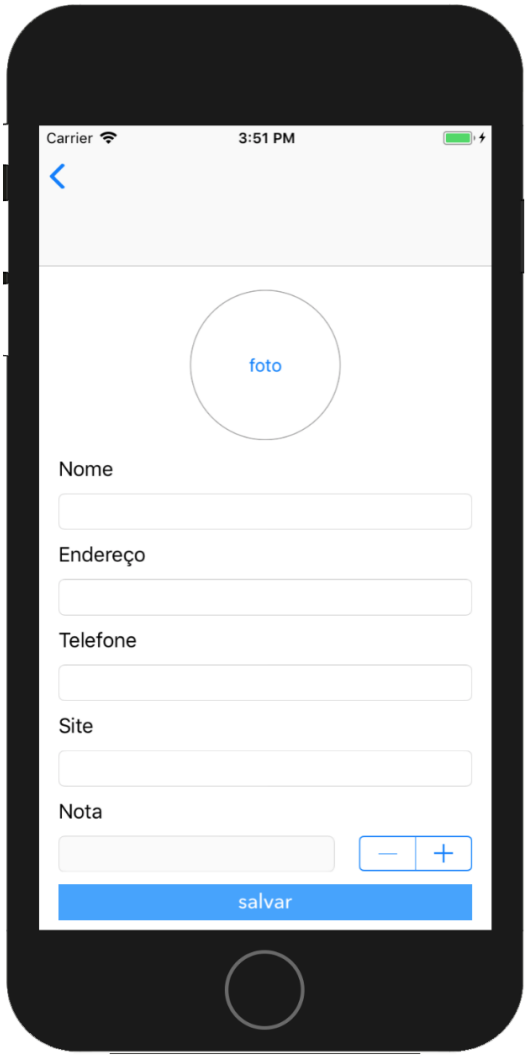
Para *começar*, vamos fazer o *download* da parte visual do projeto:

https://github.com/alura-cursos/ios-recursos-nativos-parte-1-inicio/archive/master.zip (https://github.com/alura-cursos/ios-recursos-nativos-parte-1-inicio/archive/master.zip)

Após concluir o download, vamos abrir e rodar o projeto:



A primeira feature que vamos implementar é o cadastro de alunos. Vamos clicar no botão "+" para conferir o formulário:



Aqui temos um formulário, onde vamos cadastrar as informações dos alunos. Repare que temos alguns componentes já conhecidos como: labels, textfields, buttons.

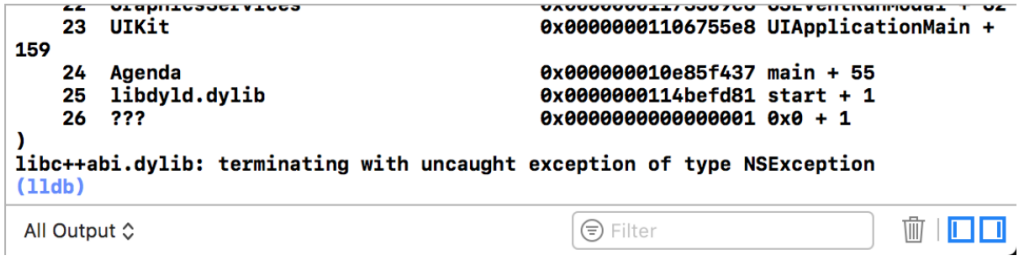
A primeira implementação que faremos será a utilização da câmera através do clique do botão "foto".

Para ter acesso as funcionalidades da câmera ou da biblioteca de fotos, utilizaremos a classe: **UIImagePickerController**.

```
@IBAction func buttonFoto(_ sender: UIButton) {
        let multimidia = UIImagePickerController()
        multimidia.sourceType = .camera
        self.present(multimidia, animated: true, completion: nil)
}
```

Repare que dessa forma que implementamos, estamos pedindo para o viewController apresentar o `gerenciadorDeMultimidia`

Vamos rodar o aplicativo para testar:



Ocorreu uma exception ao clicar no botão *foto*.

Isso ocorreu porque o simulador do iOS não possui todas as funcionalidade que um iPhone/iPad de verdade tem, como por exemplo a câmera.

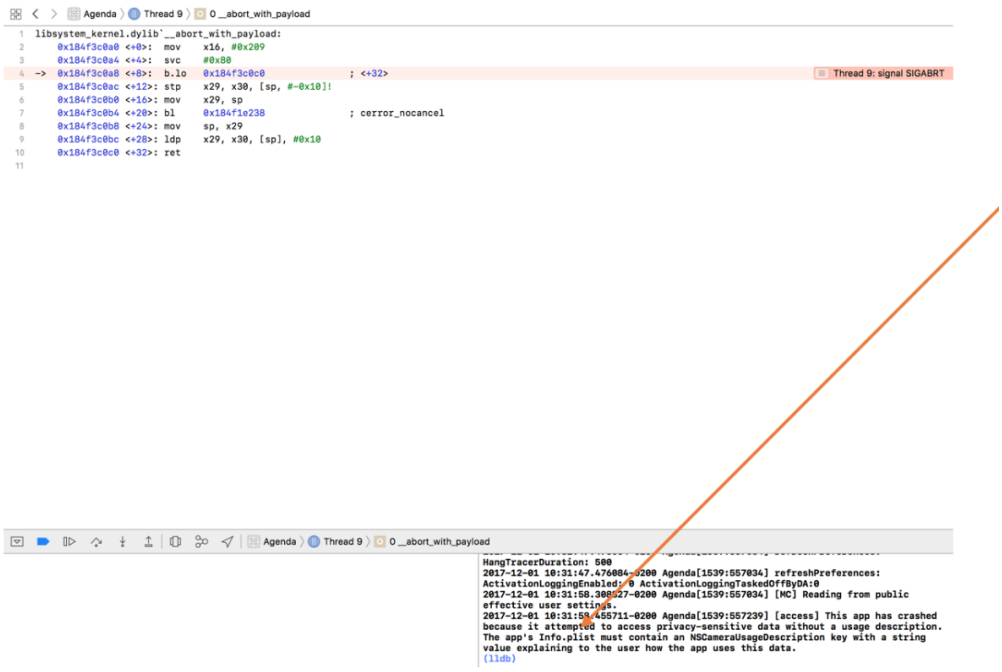
Precisamos fazer uma verificação para saber se o iOS consegue abrir a câmera ou não:

```
@IBAction func buttonFoto(_ sender: UIButton) {
        if UIImagePickerController.isSourceTypeAvailable(.camera) {
            let multimidia = UIImagePickerController()
            multimidia.sourceType = .camera
            self.present(multimidia, animated: true, completion: nil)
        }
}
```

Agora não ocorrerá erro ao clicar no botão *foto*.

Como acabamos de aprender, não é possível utilizar esse recurso no simulador. Portanto, precisaremos testar em um device de verdade.

Vamos testar:



Repare que o app deu crash novamente. Isso ocorre por questões de segurança.

O iOS exige que o usuário autorize ou não a utilização desse recurso no nosso aplicativo.

Agora é a sua vez:

- Implemente a permissão para acessar a câmera no arquivo **Info.plist**