

View Code em Swift

Como colocar elementos de UI no seu aplicativo iOS sem usar Storyboard

Com o **Storyboard** você tem a praticidade de apenas “puxar” os outlets até o código, configurar os "identifiers" e ajustar os elementos de interface.

Isso funciona muito bem se você está criando um projeto pessoal ou começando um com uma equipe que prefira essa ferramenta.

Entretanto, se você for executar a manutenção de um aplicativo que não possua Storyboard, como você vai dar assistência?

Agora vou mostrar neste artigo como fazer um aplicativo 100% **View Code**, isto é, tudo será feito programaticamente.

O código possui uma vantagem primordial quando estamos falando de projetos em que se trabalham muitas pessoas: com **View Code**, *"o código diminui drasticamente conflitos de "merge" no Git em comparação ao Storyboard"*.

Criação de um projeto do Xcode:

Crie normalmente um novo projeto com nome, time e nome da organização.

Choose options for your new project:

Product Name:

Team:

Organization Identifier:

Bundle Identifier:

Interface:

Language:

☐ Use Core Data

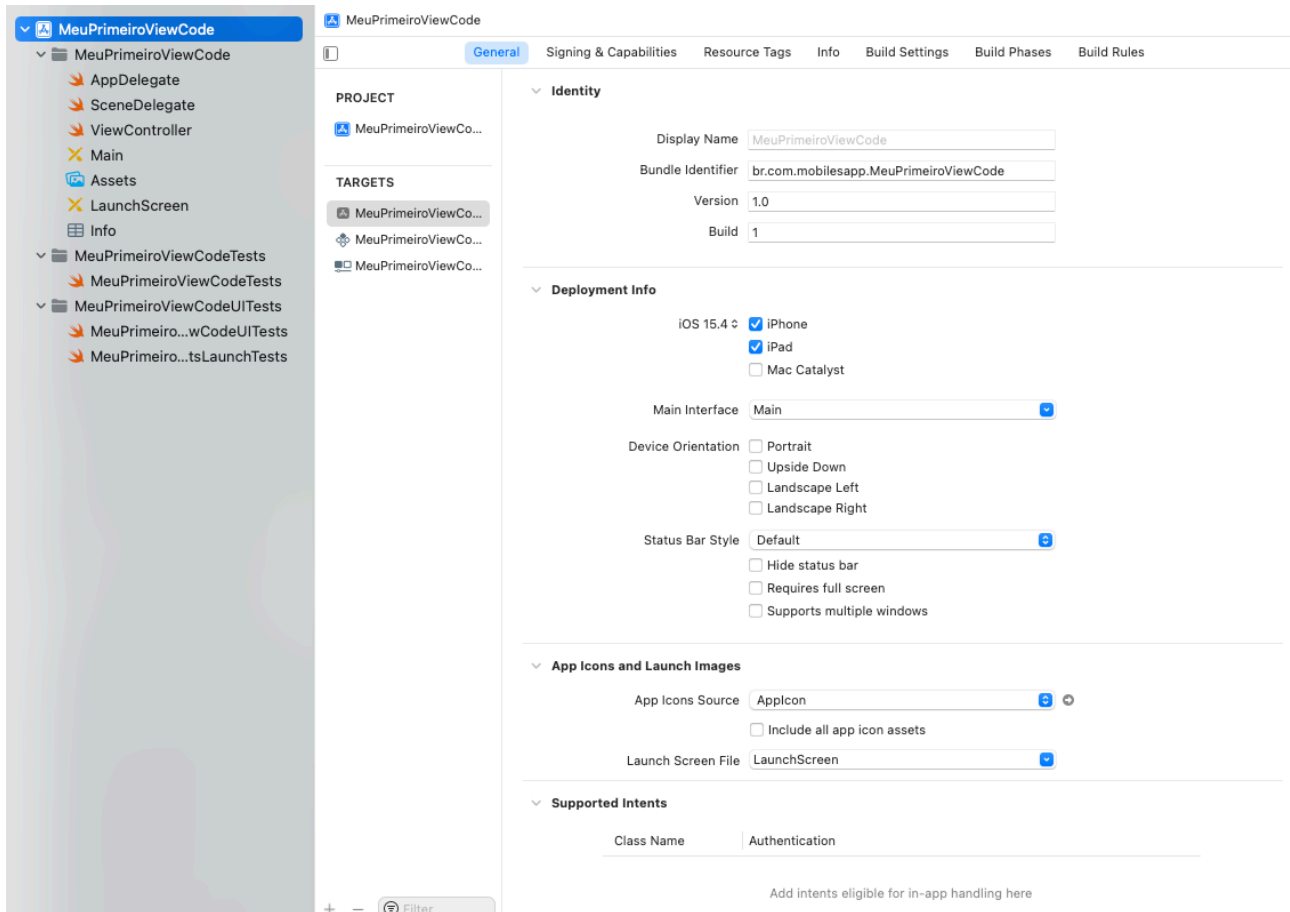
☐ Host in CloudKit

☒ Include Tests

Estou utilizando o xCode 13.3



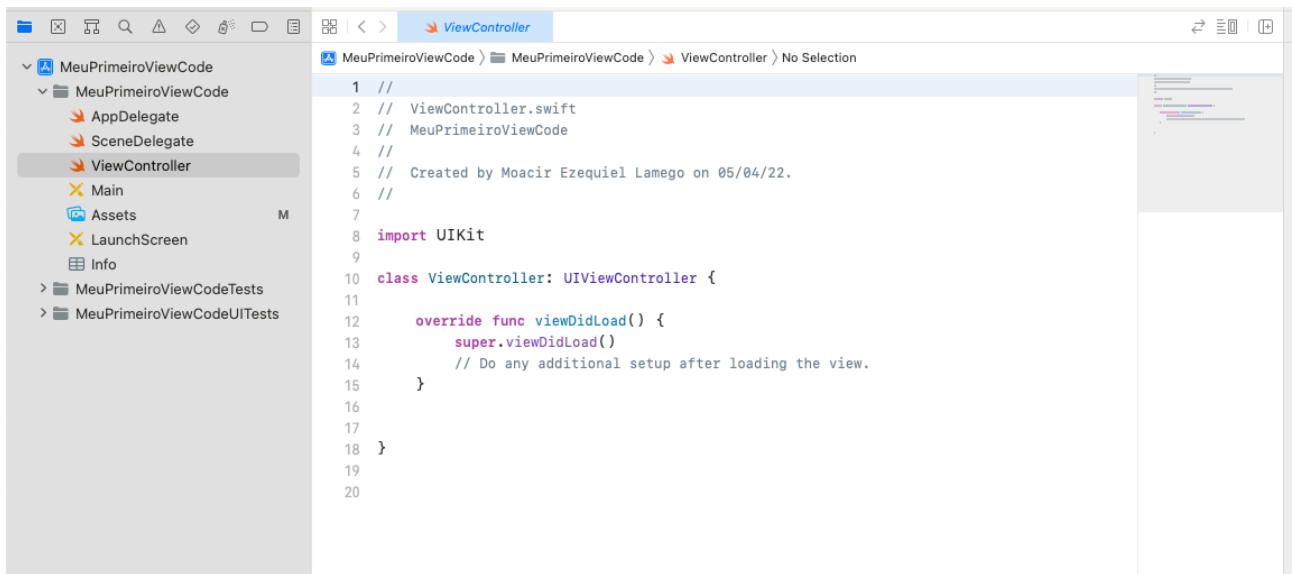
Neste exemplo **não irei mostrar** como criar um projeto **100% View Code**, mas você vai poder encontrar ele no Curso **Criando um Aplicativo iOS do zero até a publicação na Apple Store**



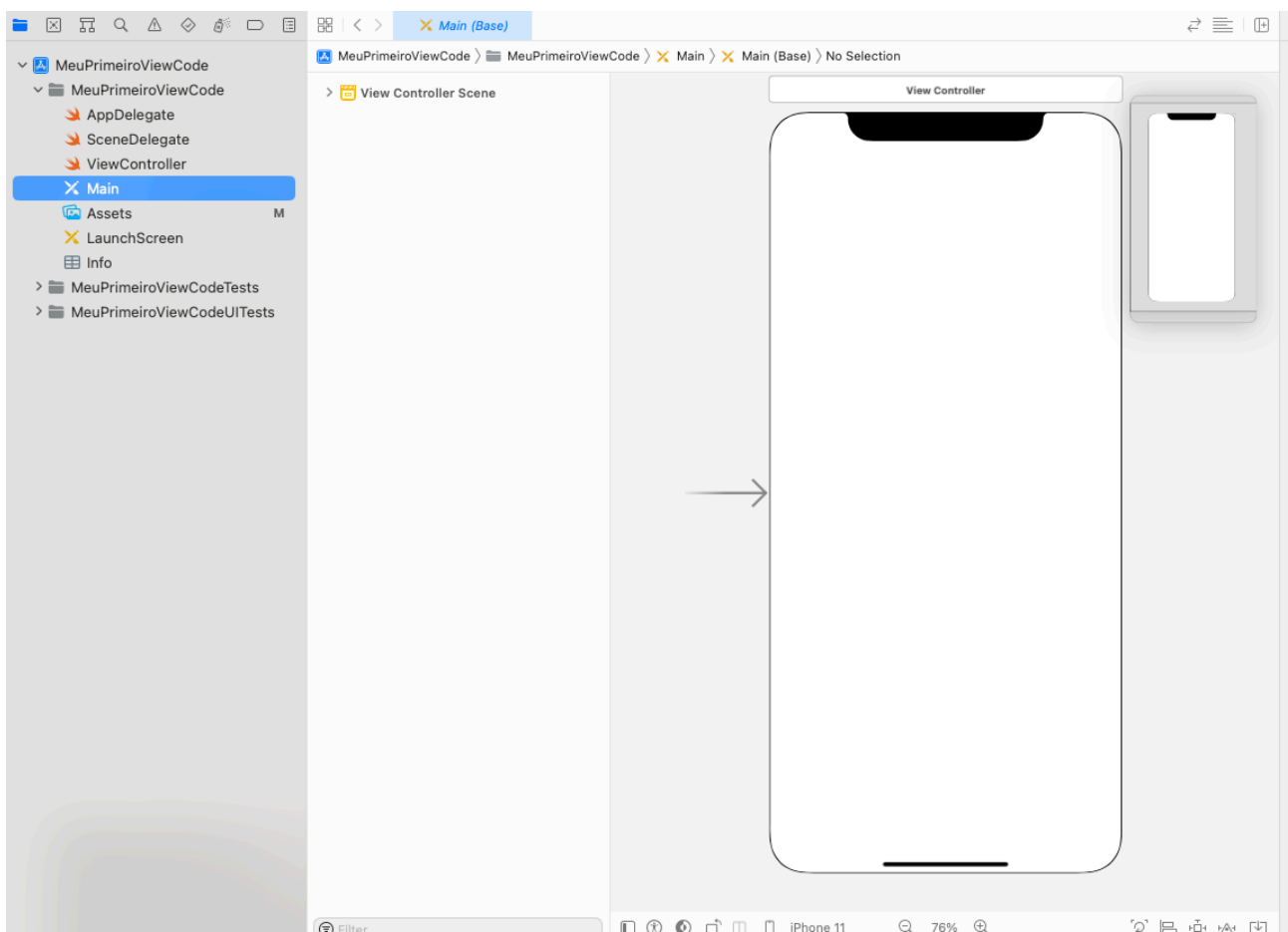
Você deve estar vendo uma tela parecida com esta agora, observe que você tem um os arquivos :

AppDelegate	Um conjunto de métodos para gerenciar comportamentos compartilhados para seu aplicativo.
SceneDelegate	Gerencie várias instâncias da IU do seu aplicativo simultaneamente e direcione os recursos para a instância apropriada da sua IU.
ViewController	O controlador de exibição associado a este item. Neste caso nosso Storyboard (Main) tem uma tela vazia que esta ligada a este controlador.

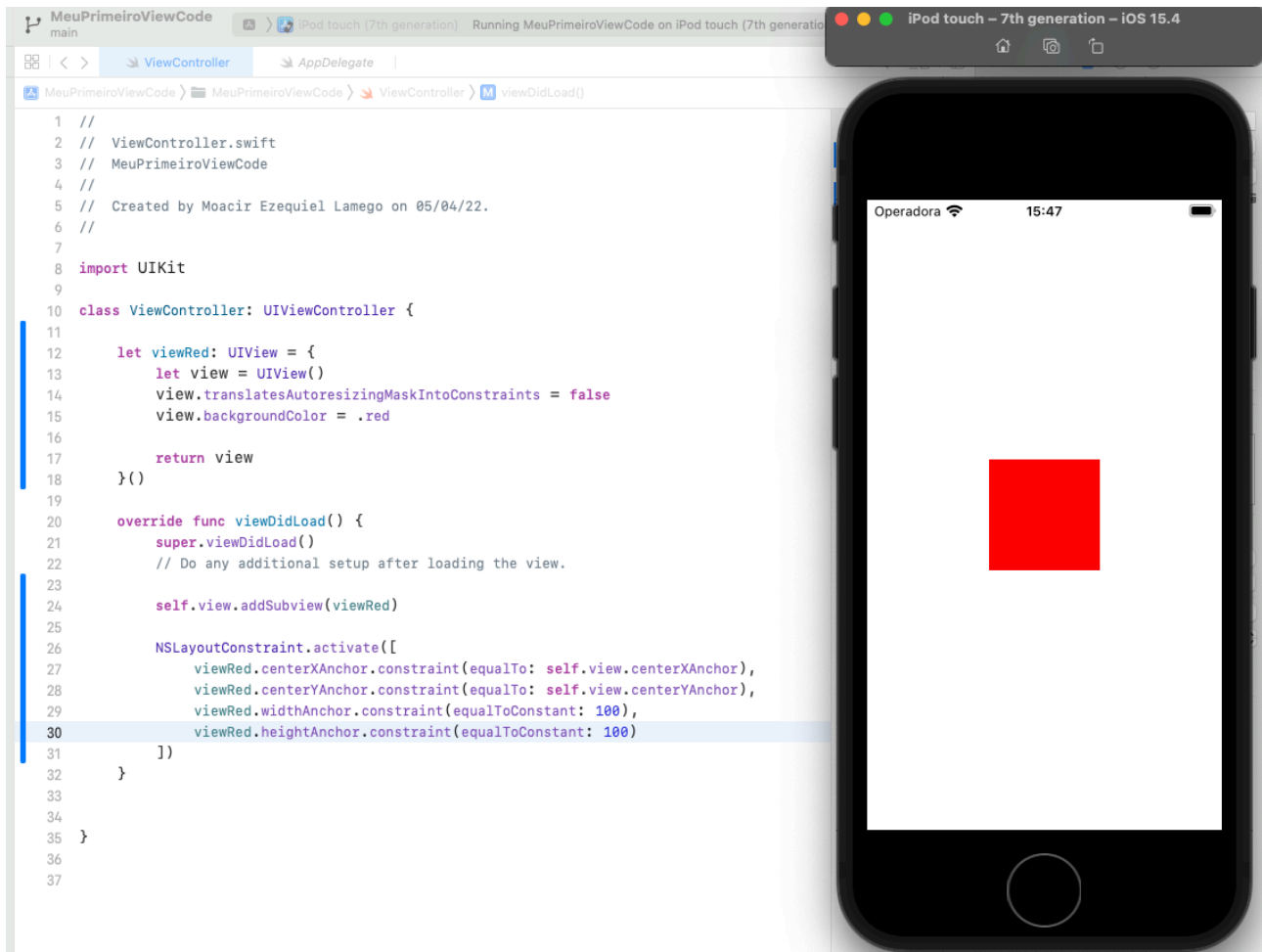
Main	Storyboard criado inicialmente com o projeto que contem apenas uma ViewController apontada para nossa classe ViewController.
Assets	Biblioteca de mídia de um usuário.
LaunchScreen	Storyboard com uma unica View que será sempre exibida ao abrir o aplicativo.
Info	Arquivo de configuração que contem configurações especifica deste aplicativo.



Como você pode conferir os arquivos estão vazios.



Vamos começar nossa brincadeira, vamos incluir no meio da tela do App uma View vermelha.



Na linha 12 criei uma UIView com o nome de viewRed, utilizando uma técnica chamada de Closures (veja o artigo Closures)

```
let viewRed: UIView = {  
    let view = UIView()  
    view.translatesAutoresizingMaskIntoConstraints = false  
    view.backgroundColor = .red  
  
    return view  
}()
```

"view.translatesAutoresizingMaskIntoConstraints = false" é um propriedade booleano que determina se a máscara de dimensionamento automático da exibição é convertida em restrições de layout automático. Ou seja se não estiver false as constraints não vão funcionar, aí você vai ficar tentando entender porque não está posicionando e vai descobrir que esqueceu de colocar este cara.

Na linha 24 adicionamos a viewRed na super view,

```
self.view.addSubview(viewRed)
```

Na linha 26 adicionamos as constraints na viewRed

```
NSLayoutConstraint.activate([
    viewRed.centerXAnchor.constraint(equalTo: self.view.centerXAnchor),
    viewRed.centerYAnchor.constraint(equalTo: self.view.centerYAnchor),
    viewRed.widthAnchor.constraint(equalToConstant: 100),
    viewRed.heightAnchor.constraint(equalToConstant: 100)
])
```

Viu como View Code é simples, não tem segredo, pronto agora você já é um programador View Code, boa sorte, até a próxima.

Ops.... Só que não né. Este foi só um artigo simples para quebrar o gelo.

Programar para View Code requer muito conhecimento dos componentes e suas propriedades. Então no nosso próximo Artigo vamos falar deles, vamos implementar os principais componentes e brincar um pouco mais de programar.