Modularizando seu código

Leonardo V. Kaminski Ferreira Engenheiro iOS @ Mindvalley github.com/leonardo-ferreira07 leonardo@mindvalley.com

Menos #C + #V

O que é?







Por que modularizar?



- Escrever menos código
- Códigos mais limpos

Modularização

- Grande escala
- Pequena escala

Modularização - Pequena escala

- Métodos
- Classes
- Struct's

Modularização - Pequena escala

```
// MARK: - Actions
72
73
\odot
       @IBAction func pickImageFromLibraryClicked(_ sender: UIBarButtonItem) {
            let imagePicker = UIImagePickerController()
75
            imagePicker.delegate = self
76
            imagePicker.allowsEditing = true
77
            imagePicker.sourceType = .photoLibrary
78
            present(imagePicker, animated: true, completion: nil)
79
        }
80
81
\odot
        @IBAction func pickImageFromCameraClicked(_ sender: UIBarButtonItem) {
            let imagePicker = UIImagePickerController()
83
            imagePicker.delegate = self
            imagePicker.allowsEditing = true
            imagePicker.sourceType = .camera
86
            present(imagePicker, animated: true, completion: nil)
87
        }
88
89
```

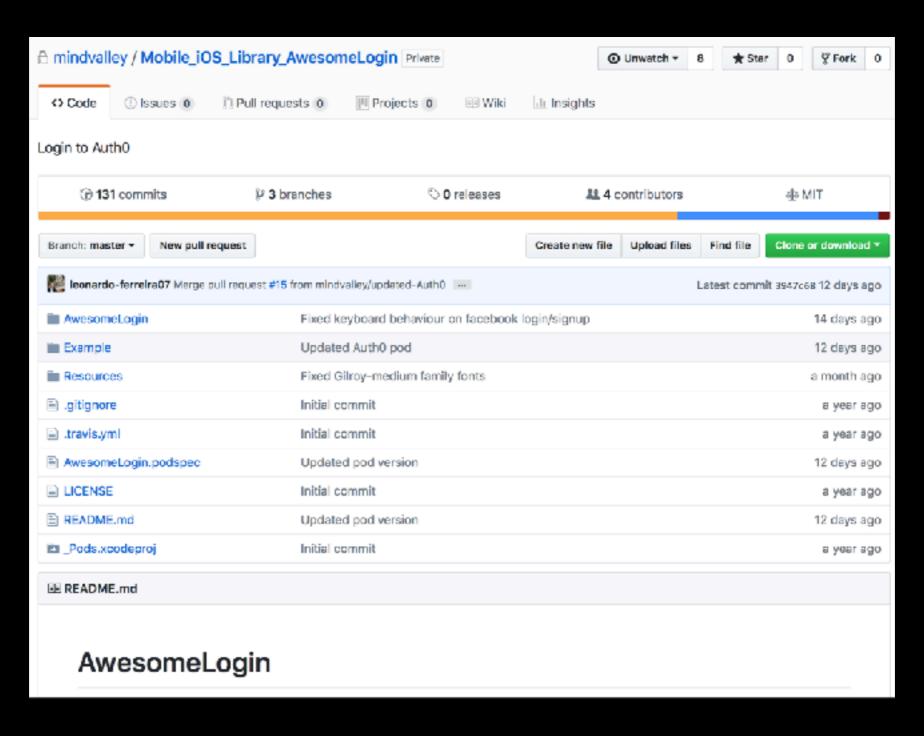
Modularização - Pequena escala

```
// MARK: - Actions
72
73
\odot
       @IBAction func pickImageFromLibraryClicked(_ sender: UIBarButtonItem) {
            chooseSourceType(sourceType: .photoLibrary)
75
       }
76
77
\odot
       @IBAction func pickImageFromCameraClicked(_ sender: UIBarButtonItem) {
            chooseSourceType(sourceType: .camera)
79
80
81
       // MARK: - ImagePicker Helper
82
83
       func chooseSourceType(sourceType: UIImagePickerControllerSourceType) {
           let imagePicker = UIImagePickerController()
85
            imagePicker.delegate = self
86
            imagePicker.allowsEditing = true
87
           imagePicker.sourceType = sourceType
88
            present(imagePicker, animated: true, completion: nil)
89
90
```

Modularização - Grande escala

- Várias classes, struct's
- Bibliotecas (frameworks)

Modularização - Grande escala



Modularização

Vantagens e desvantagens

Vantagens

- Código limpo
- Abstraído
- Orientado à componentes
- Testes

Vantagens

- Fácil de migrar
- Isolado
- Responsável (por si só)
- Prototipagem rápida

Desvantagens

- Assustador
- Abstração de código
- Transferência de arquivos
- Desacoplamento de código

Desvantagens

- Integração da nova biblioteca ou código

Casos reais

- AwesomeLogin
- AwesomeConstants
- AwesomeUIMagic
- AwesomeMedia

Casos reais

- AwesomePurchase
- AwesomeLoading
- AwesomeCore
- AwesomeLocalization

Ao final

Obrigado!

Nós temos vagas:

careers.mindvalley.com/iosdeveloper