



Introdução ao desenvolvimento iOS Nativo usando Swift



Workshop CeSIUM 2022



github.com/9991david/workshop-iOS-CeSIUM-2022

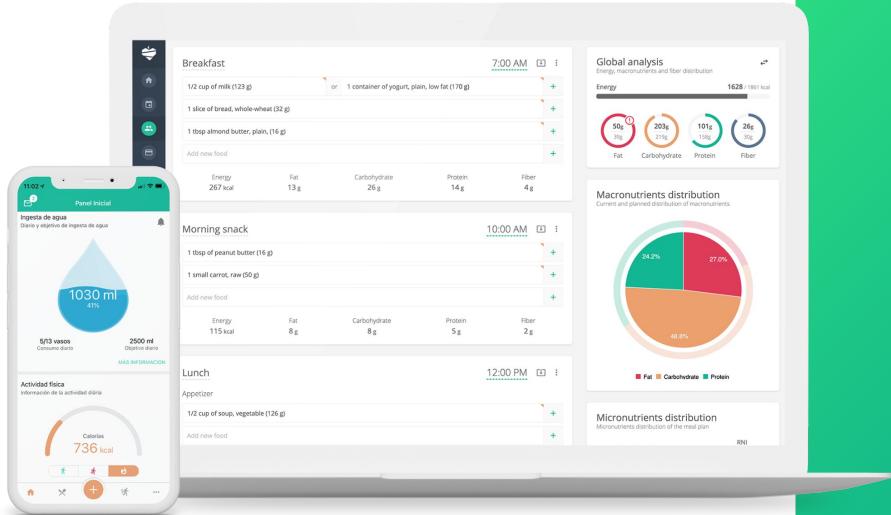
Workshop CeSIUM 2022



Agenda

- Nutrium
- Apple
- Swift
- App iOS
- Fazer uma App





PARTE 01

Nutrium

<https://nutrium.com/>

01. Nutrium

1.1 Quem somos?

- Começou na UCE15/PI/PEI por antigos alunos de Engenharia Informática
 - Atualmente somos +40 de pessoas
- É uma das referências mundiais no que toca a Software de Nutrição
 - We are all over the World!!!
- 4.9M€ em investimentos
 - 4.3M€ em Outubro de 2020

Make nutritional care accessible for everyone



01. Nutrium

1.2 Equipa Mobile

1.2.1 O que fazemos?

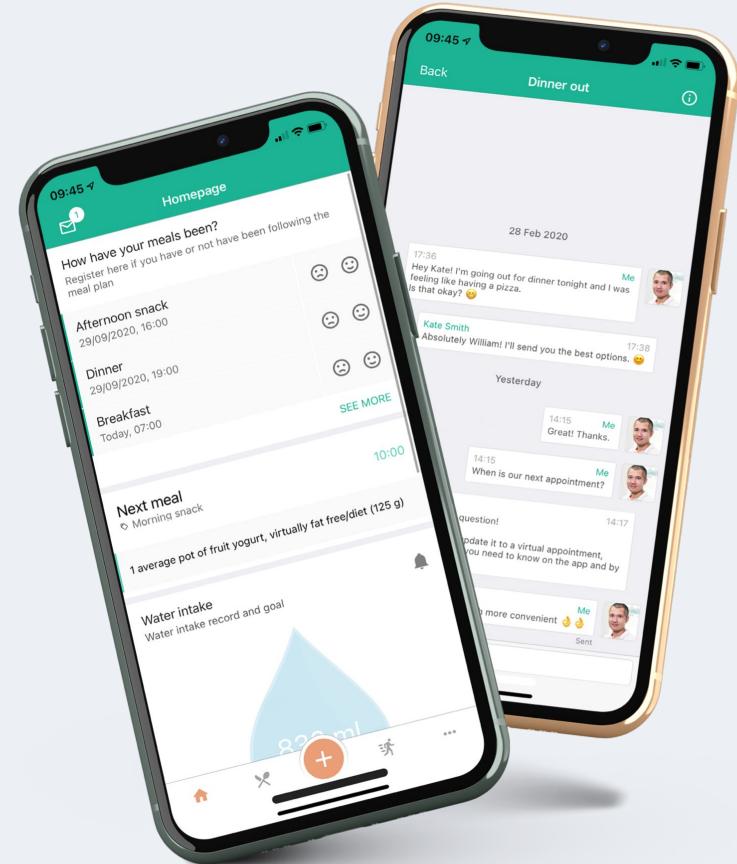
Desenvolvimento de novas funcionalidades em Android e iOS para os pacientes quer para os profissionais. Também fazemos manutenção de funcionalidades antigas e suporte aos pacientes e profissionais.

1.2.2 Como fazemos?

- Somos uma equipa de 5 pessoas - 3 Android e 2 iOS
- Usamos uma metodologia Ágil - Scrum

1.2.3 Que ferramentas usamos?

- Git
- Jira
- Slack
- CircleCI
- Xcode e Android Studio



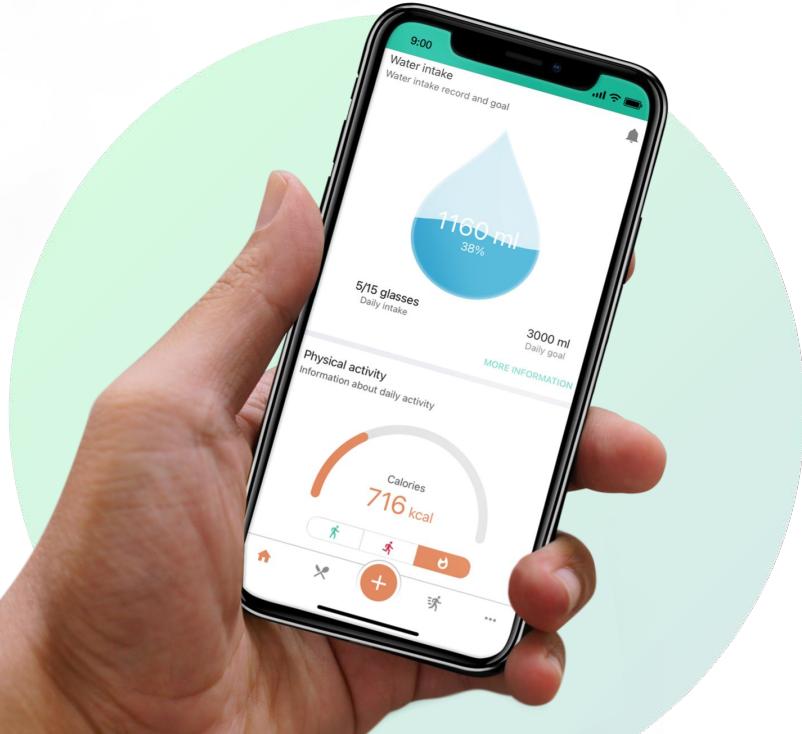


PARTE 02

Apple
www.apple.com

02. Apple

2.1 Ecossistema



- Telemóveis
- Computadores
- Relógios
- Box para televisão
- Sistema de pagamentos
- Entre outros



02. Apple

2.2 Desenvolvimento Nativo vs Híbrido



2.2.1 Nativo

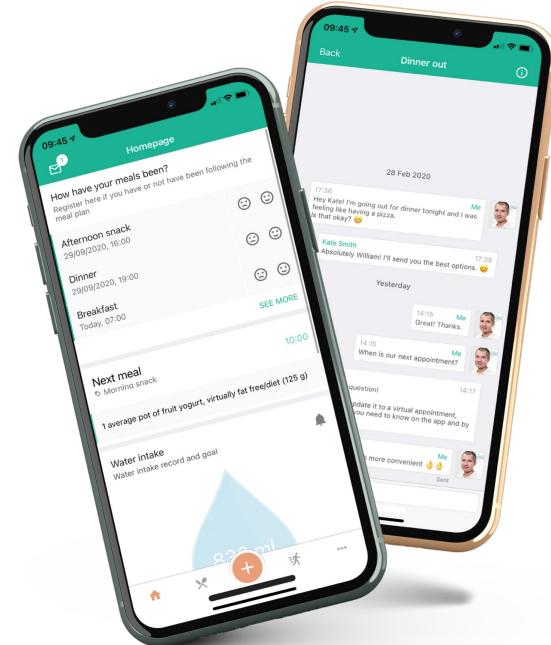
Usa-se uma linguagem e ferramentas específicas a uma plataforma (iOS ou Android)

- + Acesso a mais e melhores APIs e funcionalidades do dispositivo
- + Melhor User Experience
- + Melhor Performance
- Maior custo de desenvolvimento para diferentes plataformas
 - Tem de se saber pelo menos Objective-C/Swift e Java/Kotlin para se lançar a mesma app em iOS e Android

2.2.2 Híbrido/Cross-Platform

Geralmente, com o mesmo código consegue-se desenvolver uma mesma Aplicação para diferentes plataformas

- + Desenvolvimento mais rápido e "barato"
 - Uma só pessoa consegue fazer código para ambas as plataformas ao mesmo tempo
- + Única fonte de código para diferentes plataformas
- Pior Performance
- Acesso limitado ou mais complicado a algumas APIs e funcionalidades



02. Apple

2.3 Porquê iOS?

- Bastante procura de iOS Developers
 - Cada vez mais empresas adotam uma ideia de Mobile First
- Sempre em constante evolução/A aprender algo novo



Normalmente uma app em iOS gera mais dinheiro do que em Android





PARTE 03

Swift
www.swift.org

03. Swift

3.1 O que é?

Linguagem de programação desenvolvida pela Apple para ser usada no seu ecossistema

- É Open-Source
- Neste momento existem vários projetos para outros sistemas operativos e diferentes áreas

Encontra-se em constante evolução desde que foi anunciada em 2014

- Atualmente já se encontra na versão 5.6 (14 de março de 2022)

- <https://www.swift.org/>
- <https://docs.swift.org/swift-book/LanguageGuide/TheBasics.html>
- <https://github.com/apple/swift>



3.2 Alguns tipos de dados

```
let idade: Int = 22
let altura: Double = 1.70
let possuiCartaDeConducao: Bool = true
let nome = "Desconhecido"
let arrayDeStrings = ["string 1", "string 2"]
let dicionarioDeStrings = ["key 1": "string 1", "key 2": "string 2"]

let stringPosicaoZero = arrayDeStrings[0]
let stringDaKey1 = dicionarioDeStrings["key 1"]
```

03. Swift

3.3 Valor Mutável (var) VS Imutável (let)

3.3.1 Mutável

`var` significa que o valor pode ser alterado

```
var idade: Int = 22  
idade = 23
```

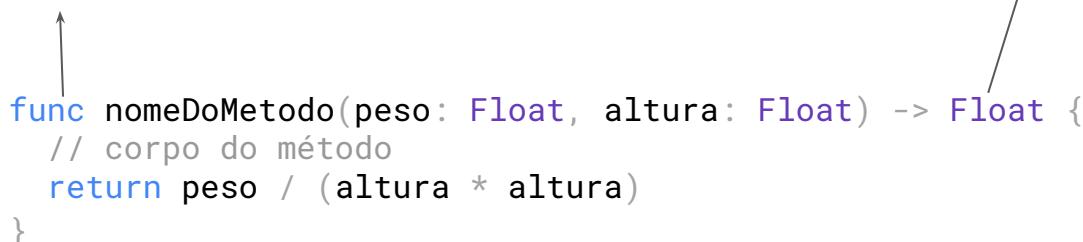
3.3.2 Imutável

`let` significa que o valor não pode ser alterado

```
let idade: Int = 22  
idade = 23 // Erro de compilação
```

3.4 Declarar um método

keyword necessária para indicar o início da função



```
func nomeDoMetodo(peso: Float, altura: Float) -> Float {  
    // corpo do método  
    return peso / (altura * altura)  
}
```

se o método não devolve nada, o return value não é necessário

```
calcularImc(peso: 65.4, altura: 1.7)
```

03. Swift

3.5 Opcionais

Um valor opcional representa um valor que pode ser **nulo**. Ao declarar-se um valor como opcional estamos a proteger o nosso código contra acessos a variáveis que são **nulas**, evitando os famosos **NullPointerException**.

```
func darOsParabens(_ nomeDaPessoa: String?) {  
    // Se o nome da pessoa for nil o valor usado vai ser "Desconhecido"  
    print("Parabéns \(nomeDaPessoa ?? "Desconhecido")")  
}  
  
darOsParabens(nil) // Imprime Desconhecido  
darOsParabens("José") // Imprime José
```

Nota: Para representar um valor nulo em swift usa-se a keyword **nil**

3.6 Controlo de Fluxo

3.6.1 Loops

- For-in
- While
- Repeat-While

```
repeat {  
    código a executar  
} while condição
```

3.6.2 Condições

- If/Else If/else
- Switch
- Guard

```
func ola(_ pessoa: [String: String]) {  
    guard let nome = pessoa["nome"] else {  
        return  
    }  
    print("Olá \\" + nome + "!")  
}
```

03. Swift

3.7 Extension

Uma extension é adicionar funcionalidades a algo já existente.

```
extension UIView {  
    func rodar() {  
        ...  
    }  
}
```

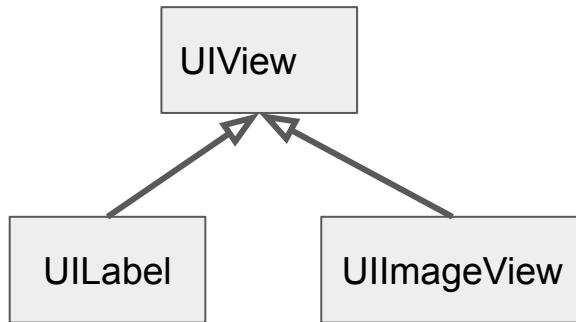
```
let view = UIView()  
view.rodar()
```

```
let label = UILabel()  
label.rodar()
```

03. Swift

3.8 Class

- É possível usar o conceito de herança
- Reference type
- Possui propriedades e métodos
- Pode adotar interfaces (Protocol)



```
class UIView {  
    var center: CGPoint  
    func draw(_ rect: CGRect) { }  
}  
  
class UILabel: UIView {  
    override func draw(_ rect: CGRect) { ... }  
}  
  
class UIImageView: UIView {  
    override func draw(_ rect: CGRect) { ... }  
}
```

03. Swift

3.9 Struct

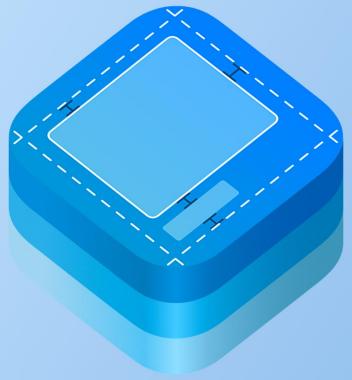
- Não é possível usar o conceito de herança
- Value type
- Possui propriedades e métodos
- Pode adotar interfaces (Protocol)

```
struct Pessoa {  
    var nome: String  
    let dataDeNascimento: Date  
  
    init(nome: String, dataDeNascimento: Date) {  
        self.nome = nome  
        self.dataDeNascimento = dataDeNascimento  
    }  
}
```

3.10 Notas

Existe muito mais a explorar

- Gerir Erros (try/catch)
- Closures
- Enums
- Coleções
- Init/Deinit
- Gestão de memória
- Genéricos
- Operadores
- Controlo de acesso
- ...

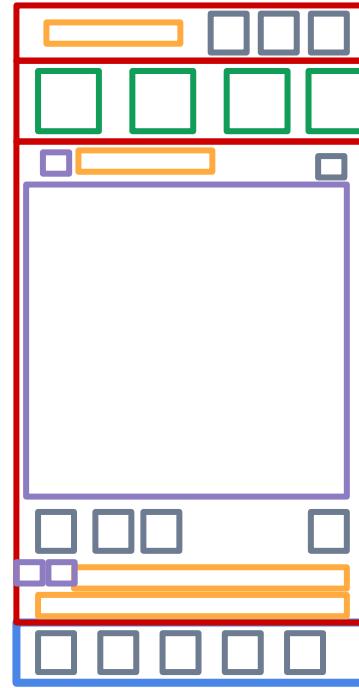


PARTE 04

App iOS

04. App iOS

4.1 Exemplos de apps



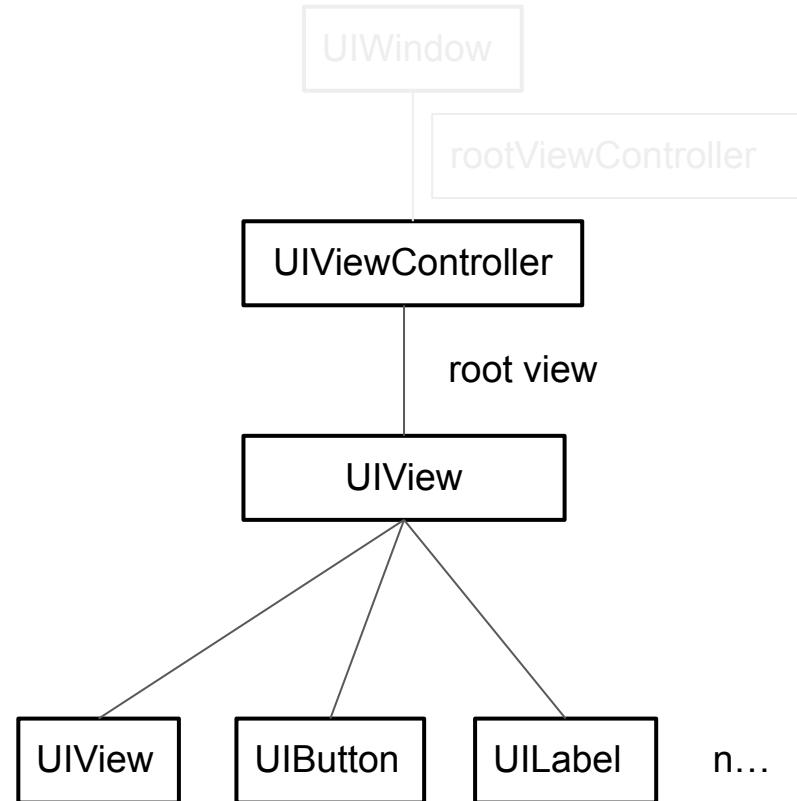
04. App iOS

4.2 UIViewController

A Apple recomenda uma abordagem Model View Controller para “gerir” a lógica da aplicação e a interação do utilizador.

Obrigatoriamente existe sempre um UIViewController na aplicação chamado de rootViewController.

Um UIViewController pode ser um container para outros viewcontrollers, exemplo disso será o UINavigationController ou até podemos fazer os nossos UIViewControllers.



- <https://developer.apple.com/documentation/uikit/uiviewcontroller>
- <https://developer.apple.com/library/archive/documentation/General/Conceptual/CocoaEncyclopedia/Model-View-Controller/Model-View-Controller.html>

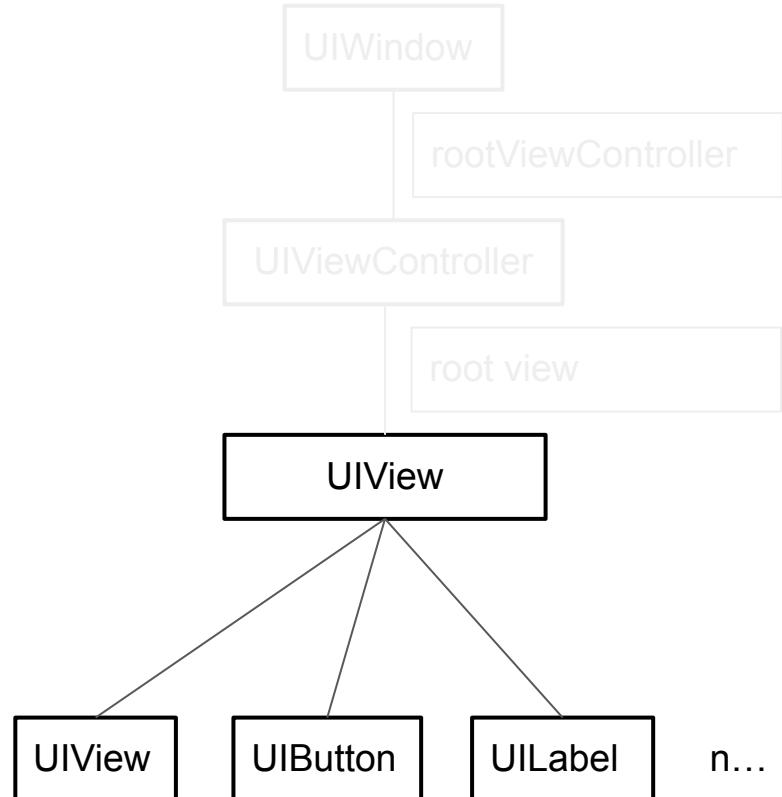
04. App iOS

4.3 UIView

Elemento base para todos os componentes que aparecem no ecrã.

Permite desenhar, fazer animações, conter outras UIView, posicionar subviews

- <https://developer.apple.com/documentation/uikit/uiview>



4.4 Alguns elementos

- UILabel
- UIButton
- UIImageView
- UITextField
- UIScrollView



Olá Mundo!

Press me!



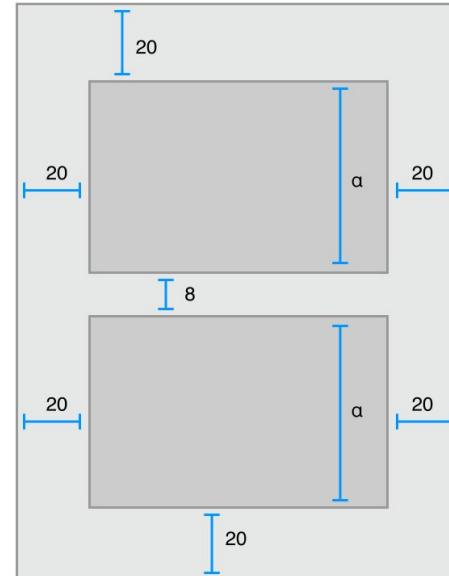
04. App iOS

4.5 Auto Layout

O sistema Auto Layout calcula o tamanho e posição de todas as views na sua hierarquia, com base nas constraints adicionadas.

Uma constraint é uma regra.

Nota: Uma view necessita de constraints que posicionem a view verticalmente e horizontalmente, no caso de faltar alguma constraint o sistema não vai conseguir posicionar a view corretamente.

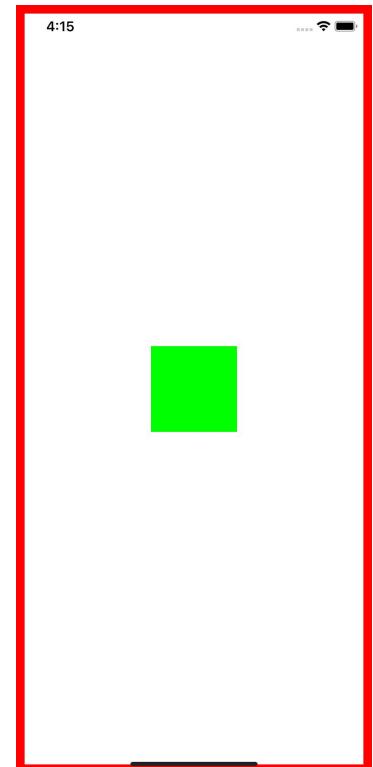


- <https://developer.apple.com/library/archive/documentation/UserExperience/Conceptual/AutolayoutPG/index.html>

04. App iOS

4.5 Auto Layout

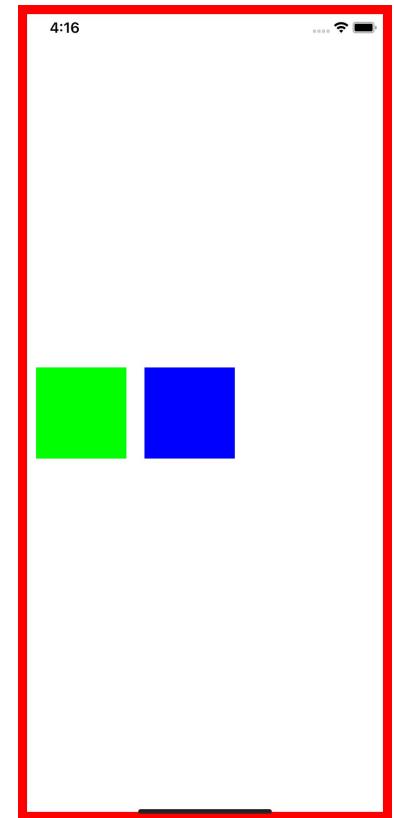
```
viewGreen.centerXAnchor.constraint(equalTo: parent.centerXAnchor),  
viewGreen.centerYAnchor.constraint(equalTo: parent.centerYAnchor)
```



04. App iOS

4.5 Auto Layout

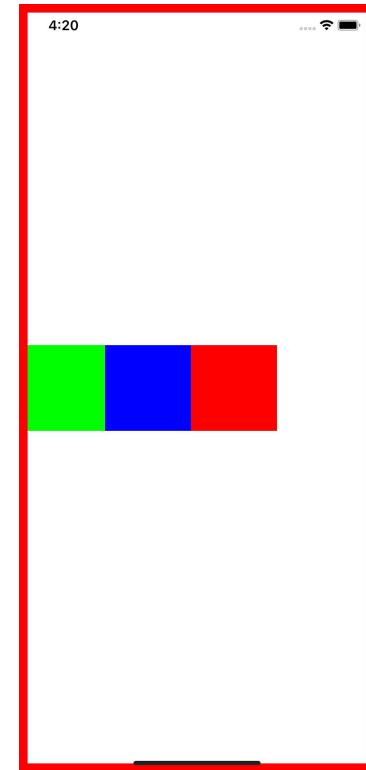
```
viewBlue.leftAnchor.constraint(equalTo: viewGreen.rightAnchor, constant: 20),  
viewBlue.topAnchor.constraint(equalTo: viewGreen.topAnchor),
```



04. App iOS

4.6 UIStackView

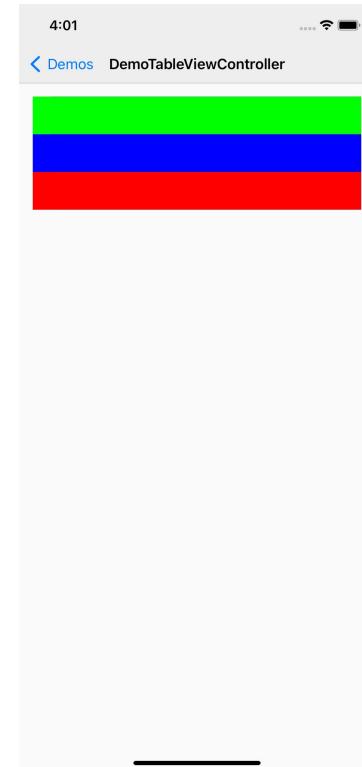
- Posiciona o conteúdo de forma vertical ou horizontal
- É posicionado usando constraints



04. App iOS

4.7 UITableView

- Posiciona o conteúdo de forma vertical
- Utiliza a largura do ecrã
- Possibilidade de adicionar ações ao fazer swipe nas células



Tópico importante: Reutilização de células

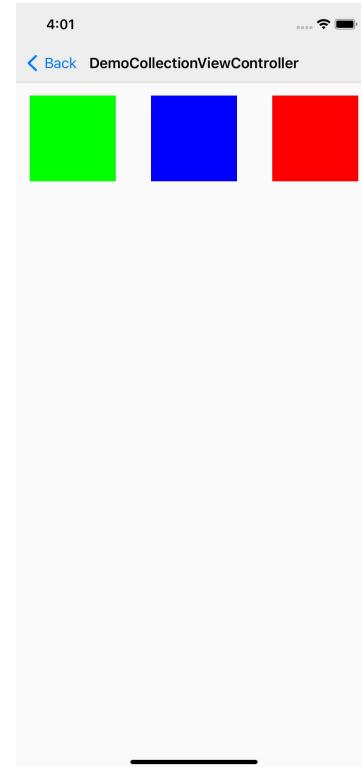
- <https://developer.apple.com/documentation/uikit/uitableview/1614891-dequeueReusableCell>

04. App iOS

4.8 UICollectionView

Possibilidade de posicionar o conteúdo de formas diferentes:

- em grelha,
- verticalmente,
- horizontalmente,
- entre muitas outras porque permite fazer o cálculo da posição dos elementos



Tópico importante:

- <https://developer.apple.com/documentation/uikit/uicollectionview/1618063-dequeueReusableCell>



PARTE 05

Fazer uma App

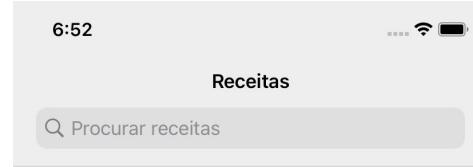
05. Fazer uma App

5.2 O que vamos criar

Aplicação para listar receitas

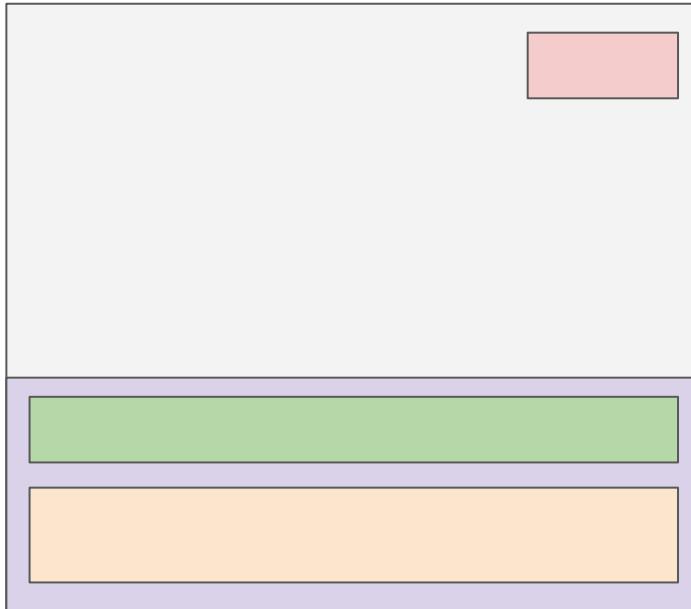
- Permite pesquisar pelo nome
- Permite ver:
 - a foto
 - o nome
 - energia
 - gorduras
 - açúcar
 - tempo de preparação
 - indicação da dificuldade
 - verde - fácil
 - laranja - média
 - vermelho - difícil

Nota: As receitas são aleatórias e podem não fazer sentido nenhum



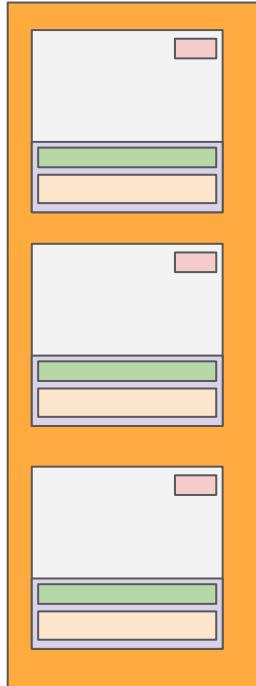
05. Fazer uma App

5.3 RecipeView



05. Fazer uma App

5.4 RecipesListViewController



05. Fazer uma App

5.5 Pesquisar Receita

Vamos usar UISearchController fornecido pela Apple

```
// UISearchResultsUpdating  
func updateSearchResults(for searchController: UISearchController)
```



- <https://developer.apple.com/documentation/uikit/uisearchcontroller>

05. Fazer uma App

5.6 Ler receitas de um ficheiro

5.6.1 JSON

As nossas receitas encontram-se num ficheiro JSON.

5.6.2 Decodable

Quando temos um tipo de dados que pode ser representado em algum formato, no nosso caso em JSON

5.6.3 JSONDecoder

O JSONDecoder é um objeto fornecido pela Apple que vai fazer o parse do ficheiro JSON para o objeto Decodable correspondente

```
{  
    "nutritionalSummary": {  
        "fat": "2,1 g",  
        "sugar": "19,0 g",  
        "energy": "311 Kcal"  
    },  
    "ingredients": [ "Corn-flakes", "Acucar baunilhado" ],  
    "id": "3B49BA5B-16A2-4ABA-BE11-CA4AAC30149",  
    "steps": [ "Step 1", "Step 2" ],  
    "isFavourite": false,  
    "image": "BASE64",  
    "difficulty": 3,  
    "createdAt": "3/16/22",  
    "name": "Curau de milho verde com leite condensado",  
    "preparationTime": "00:30 min"  
}
```



PARTE 06

Notas

06. Notas

6.1 Multithread

6.1.1 DispatchQueue

São usadas para abstrair a gestão de threads. Podem ser usadas para executar trabalho concorrente ou em fila de forma assíncrona.

6.1.2 Main VS Background

Criação e manipulação de elementos de UI são operações deve ser sempre feito na main thread. Os objetos de UI não são thread safe, ao executar sempre na Main Thread não existe problemas de data race.

Operações mais pesadas devem ser executadas na Background Thread para não bloquear a Main Thread que deve estar sempre disponível para responder aos eventos do utilizador.

- <https://developer.apple.com/documentation/dispatch/dispatchqueue>

06. Notas

6.2 Obter Receitas da internet

6.2.1 URLSession

URLSession é uma classe disponibilizada pela Apple para fazer pedidos HTTP.
Os pedidos HTTP funcionam de forma assíncrona.

```
let url = URL(string:  
"https://raw.githubusercontent.com/9991david/workshop-iOS-CeSIUM-2022/main/iOS_empty/Workshop/AllRecipes.json")!  
  
let task = URLSession.shared.dataTask(  
    with: request,  
    completionHandler: { (data, response, error) -> Void in  
        // Fazer parse do JSON recebido da internet  
    }  
)
```

- <https://developer.apple.com/documentation/foundation/urlsession>

06. Notas

6.3 Gestor de dependências



6.3.1 O que é?

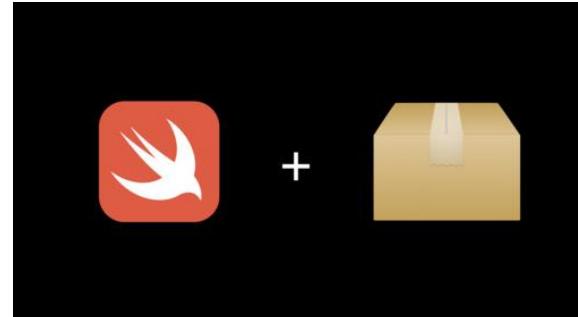
Ferramenta que ajuda a gerir dependências (aka libs externas).

Essas dependências podem ser de terceiros ou internas.



6.3.2 Exemplos

- SwiftPackageManager
 - Gerido pela Apple
 - Integrado no Xcode
 - <https://www.swift.org/package-manager/>
- CocoaPods
 - Gerido pela comunidade
 - <https://cocoapods.org/>
- Carthage
 - Gerido pela comunidade
 - <https://github.com/Carthage/Carthage>



06. Notas

6.4 Caminho a explorar

Depois de desenvolver uma aplicação ainda falta o processo de fazer o seu lançamento para a Apple Store.

- Guardar as receitas localmente numa base de dados
 - CoreData - <https://developer.apple.com/documentation/coredata>
 - Realme - <https://realm.io/realm-swift>
 - Sqlite
- Explorar o UIApplicationDelegate / UISceneDelegate
 - ciclo de vida da aplicação
- SwiftUI
 - <https://developer.apple.com/xcode/swiftui/>
- Usar o Xcode para fazer debug
- Adicionar assets
- Adicionar localizações

06. Notas

6.4 Caminho a explorar

- Reutilização de células na UITableView e UICollectionView
- Programação funcional reativa
 - Combine, oficial da Apple - <https://developer.apple.com/documentation/combine>
 - <https://github.com/ReactiveX/RxSwift>
 - <https://github.com/ReactiveCocoa/ReactiveCocoa>
 - <https://github.com/DeclarativeHub/ReactiveKit>
 - ...
- Auto Layout
 - <https://github.com/SnapKit/SnapKit>
 - <https://github.com/robb/Cartography>
 - ...
- Analytics

06. Notas

6.4 Caminho a explorar

- Icon da aplicação
- SplashScreen - ecrã que aparece ao iniciar a aplicação
 - <https://developer.apple.com/design/human-interface-guidelines/ios/visual-design/launch-screen/>
- Bundle identifier - identificador único da aplicação
 - https://developer.apple.com/documentation/appstoreconnectapi/bundle_ids
- App name
- Criar certificados de desenvolvimento e produção na conta de developer
 - preciso pagar para lançar a aplicação uma vez anualmente
 - <https://developer.apple.com>
- Analisar erros depois da app lançada na store
 - Crash reporter

06. Notas

6.5 Material Externo

- <https://developer.apple.com/swift/>
- <https://www.raywenderlich.com/>
- <https://nshipster.com/>
- <https://useyourloaf.com/>
- <https://www.swiftbysundell.com/>
- <https://academy.realm.io/section/apple/>
- <https://www.cocoawithlove.com/>
- <https://www.objc.io/>
- <https://cs193p.sites.stanford.edu/2020>
- <http://www.cimgf.com/>
- <https://iosdevweekly.com/>
- Youtube
 - <https://www.youtube.com/c/PaulHudson>
 - <https://www.youtube.com/c/PhysicsNerdDev>
 - <https://www.youtube.com/c/EssentialDeveloper>
- ...



We want you!

davidgoncalves@nutrium.com
brunoazevedo@nutrium.com