



Testando o App do Nubank

Victor Maraccini

@vmaraccini

Francesco Perrotti-Garcia

@fpg1503

Show of hands

Quem aqui:

- É cliente do Nubank

O que é Nubank?

O que é Nubank?

- Mais de 1200 funcionários
- Diversidade
- Mais de 5MM de clientes
- Clientes fazem tudo pelo app
- Cultura de qualidade e testes



Show of hands

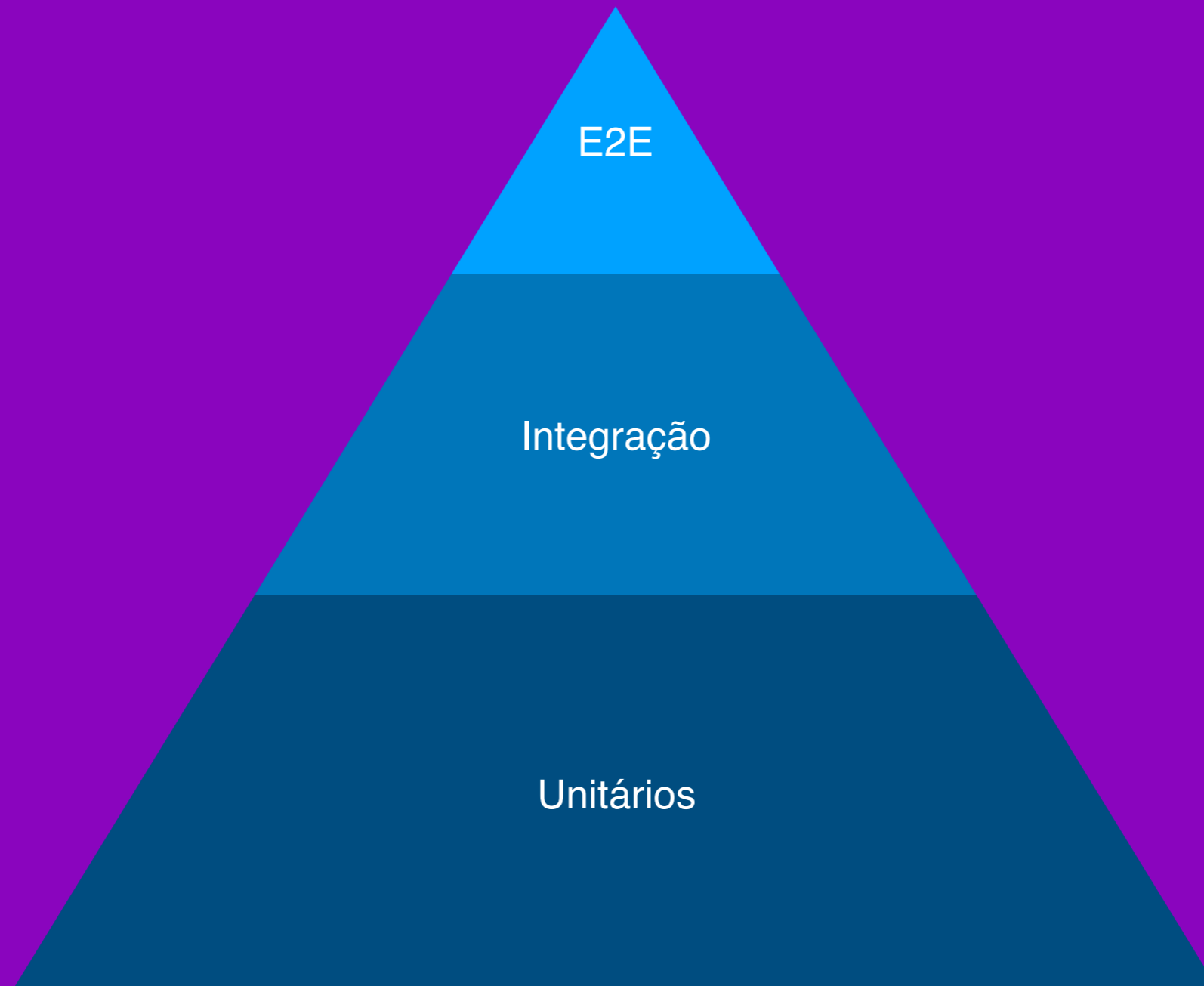
Quem aqui:

- É cliente do Nubank
- Mexeu com Mobile
- Escreveu teste para Mobile

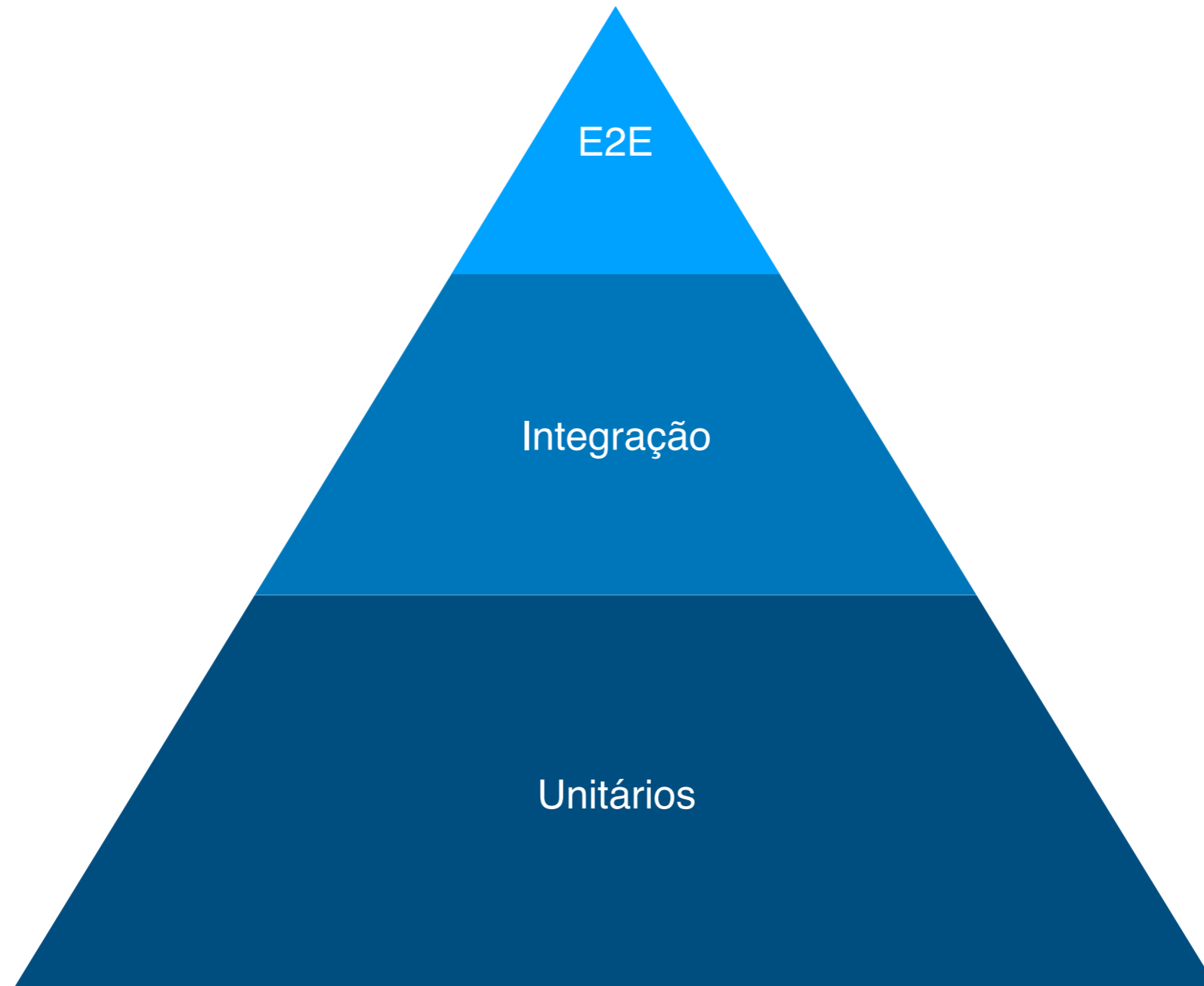
Por que testamos?

- Evitar regressão
 - Mais velocidade para refatorar e fazer mudanças
 - Novos engenheiros tem confiança nas entregas
- Garantir comportamentos existentes
 - De forma automatizada
 - Em diferentes condições de uso

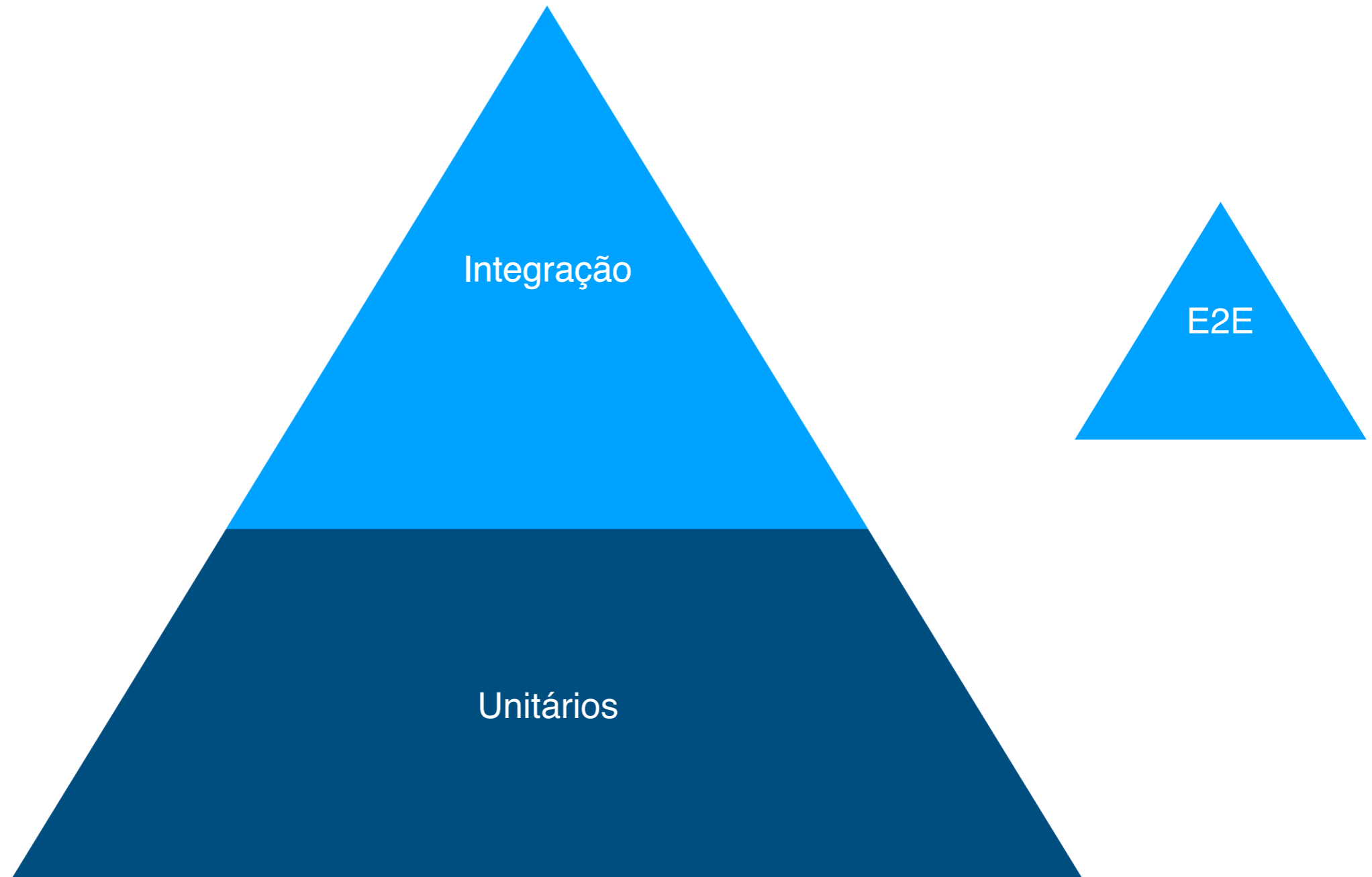
Pirâmide de testes



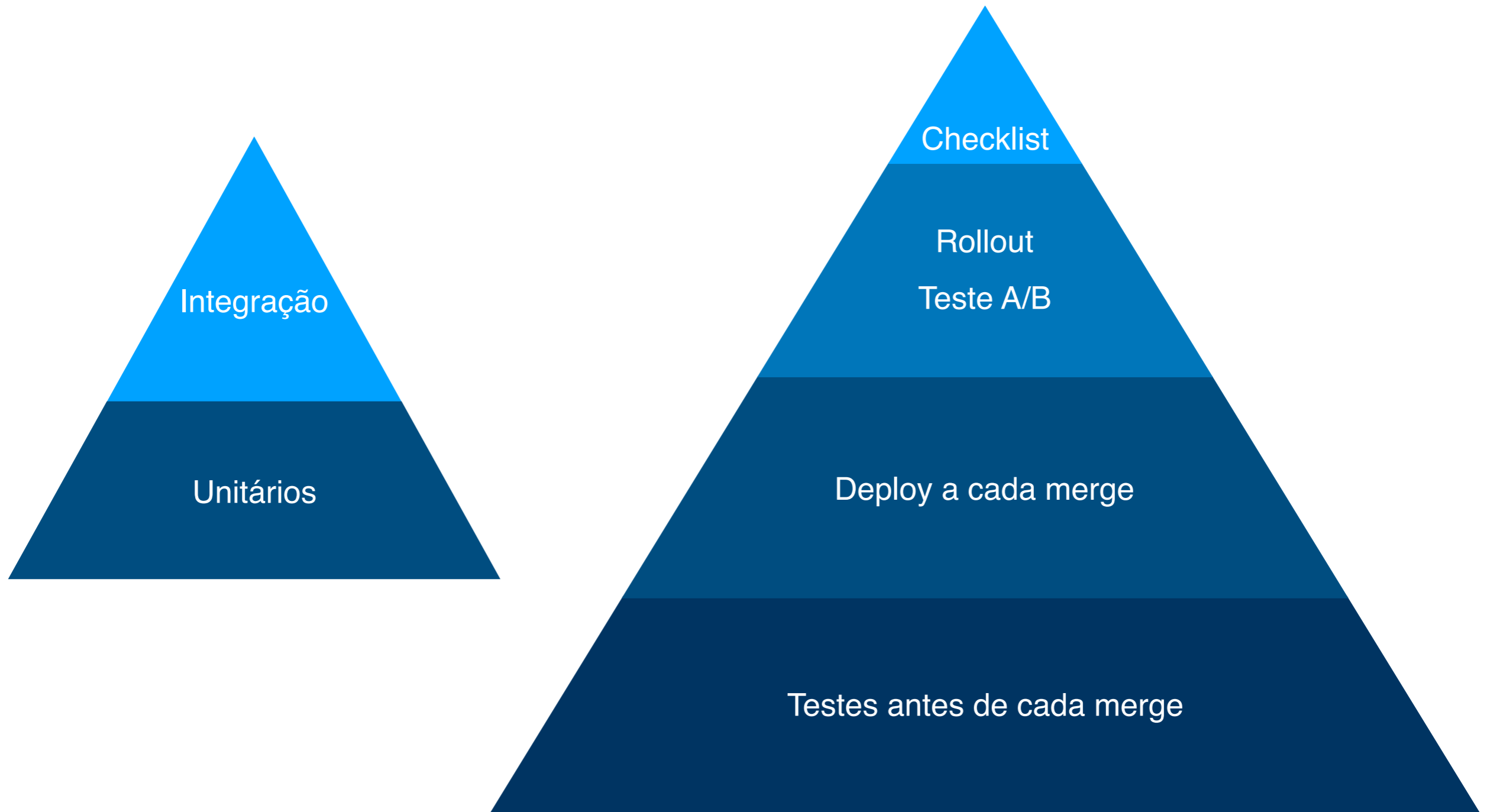
Nossa pirâmide de testes em Mobile




Nossa pirâmide de testes em Mobile









Nossa pirâmide de testes em Mobile



Integração contínua





	Code owner review required Waiting on code owner review from nubank/ios-dev. Learn more.	Show all reviewers
	All checks have passed 2 successful checks	Hide all checks
	 ios-shell/auto-merge	Details
	 ios-shell/integrate	Required Details



Code Review




Testes passando

Testes antes de cada merge

Integração contínua

- ✓  ios-shell/auto-merge
- ✓  ios-shell/integrate

ios-shell  



  

[History](#)



















Instance: 1628

[Compare](#) | [Changes](#) ▼ | [VSM](#)

Triggered by changes
on 07 Dec, 2018 at 08:10:59 Local Time

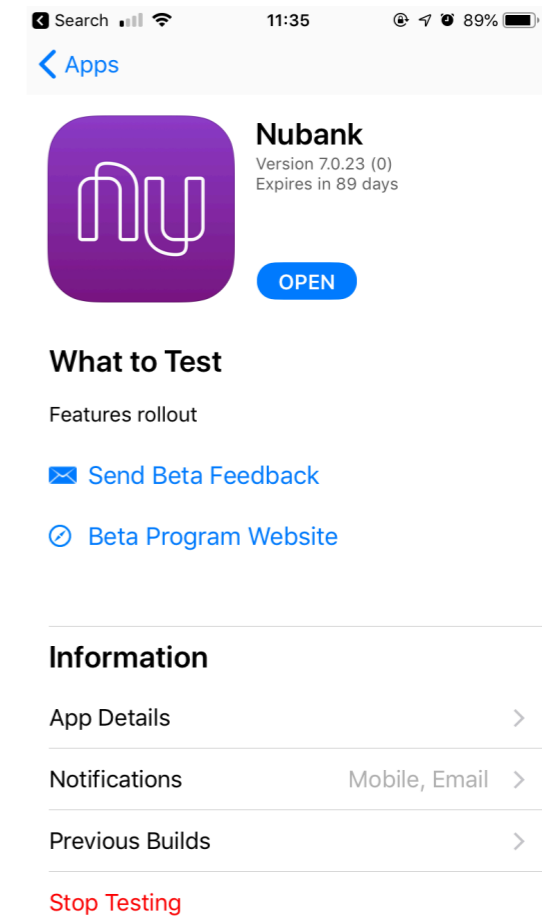
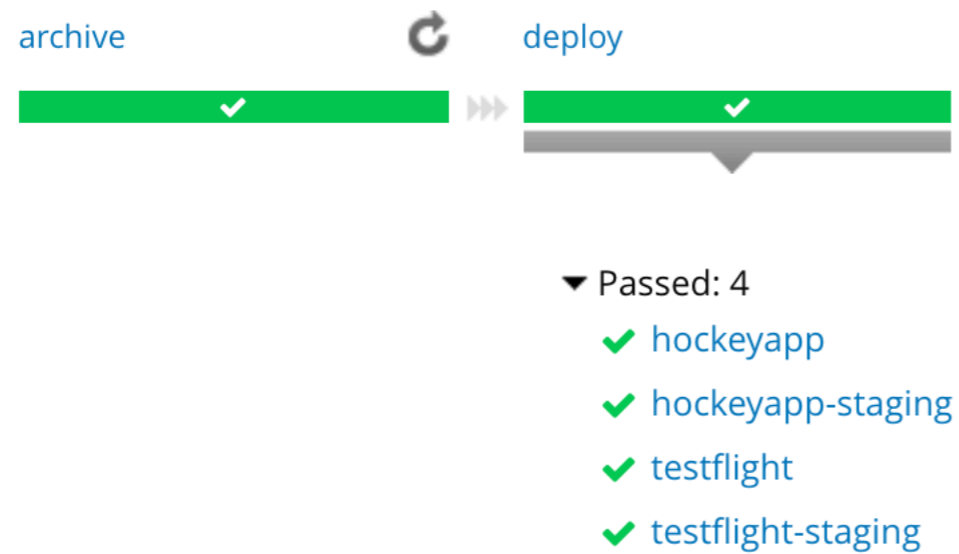
 



- ▼  MacStadium - V
- ▼  MacPro_Cluster
 -  10.92.173.13
 -  10.92.173.14
- ▼  Blue
 -  OSX-4Core-react_9_3-1zq1y
 -  OSX-4Core-react_9_3-47jnei
 -  OSX-4Core-react_9_3-l80xgj
 -  OSX-4Core-react_9_3-lkx2ze
 -  OSX-8Core-generic_9_3-8n2qev
 -  OSX-8Core-generic_9_3-9sj5e
 -  OSX-8Core-generic_9_3-h45cs9
 -  OSX-8Core-generic_9_3-j6n4es
 -  OSX-8Core-generic_9_3-tv87x2
 -  OSX-8Core-generic_9_3-xf42zk
 -  OSX-8Core-generic_9_3_jenkins-8pg17
 -  OSX-8Core-generic_9_3_jenkins-nrz3ja
 -  OSX-8Core-generic_9_3_jenkins-pmlv3o

Testes antes de cada merge

Entrega continua



Deploy a cada merge

Testes unitários

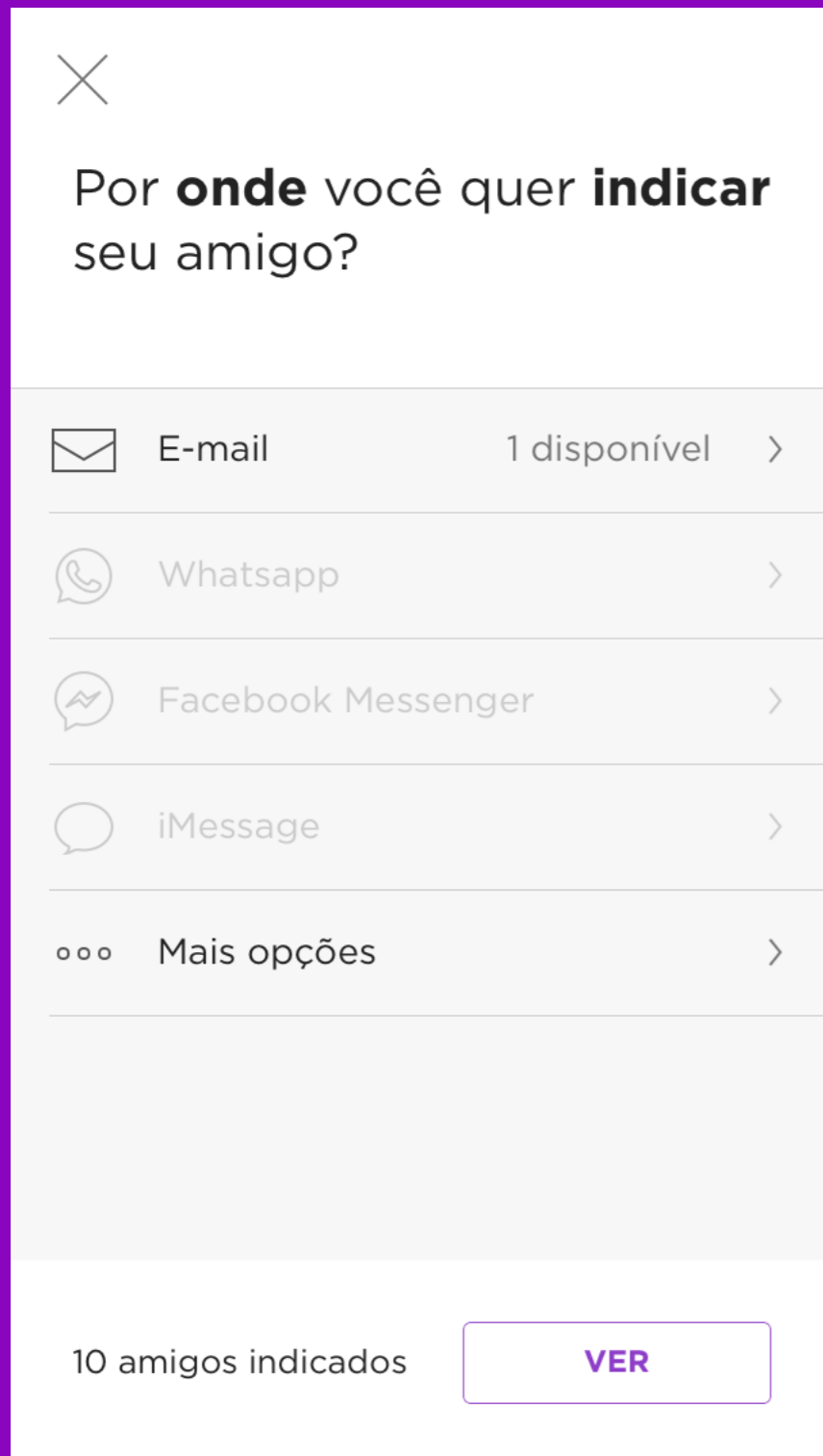


- Foco no comportamento individual de componentes
- Desacoplado
- Especializado

Testes unitários

Integração

Unitários

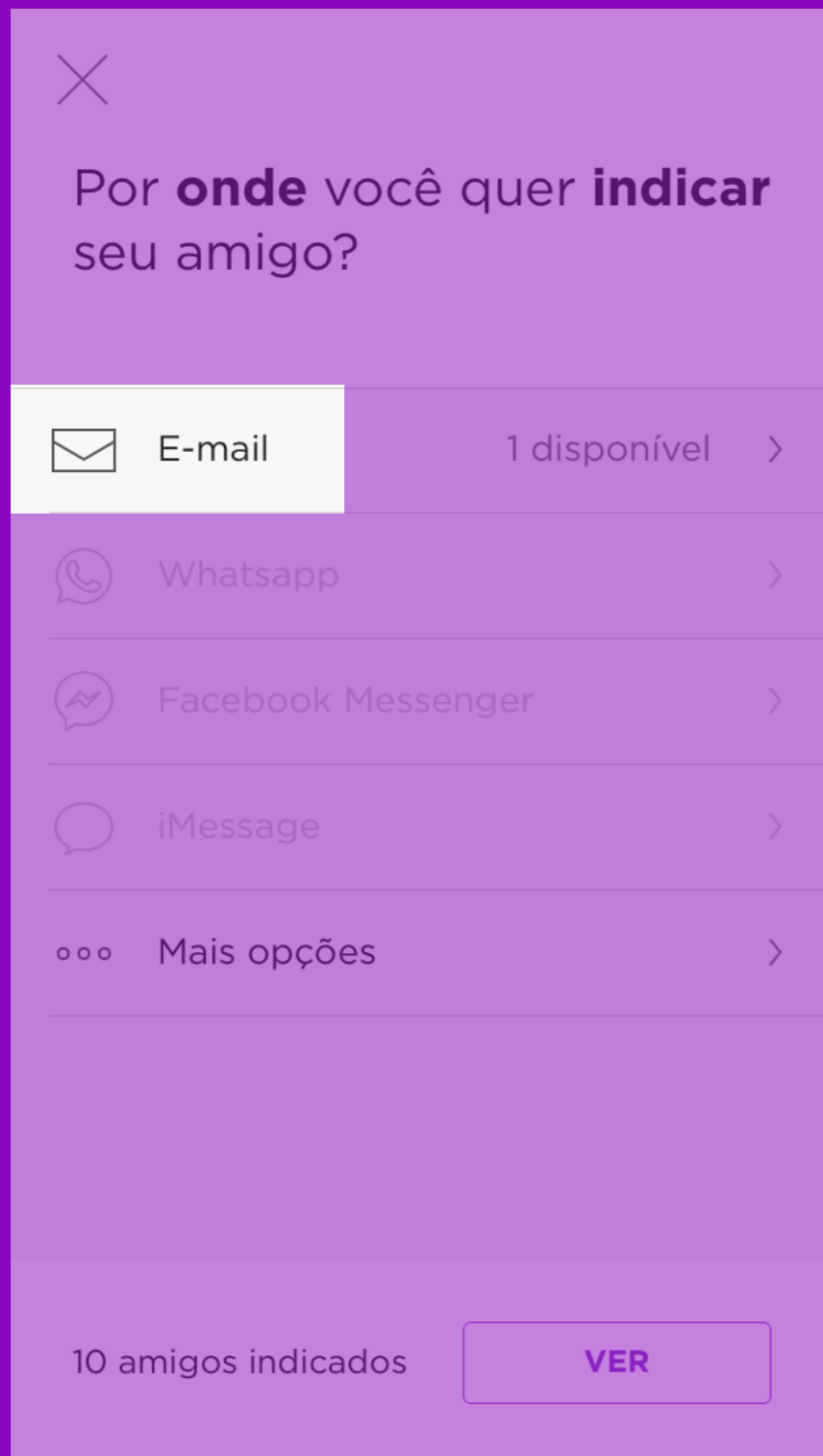


```
struct MGMChannelData {  
    let type: MGMChannel  
    let count: Int?  
    let canOpen: Bool  
}
```

Testes unitários

Integração

Unitários

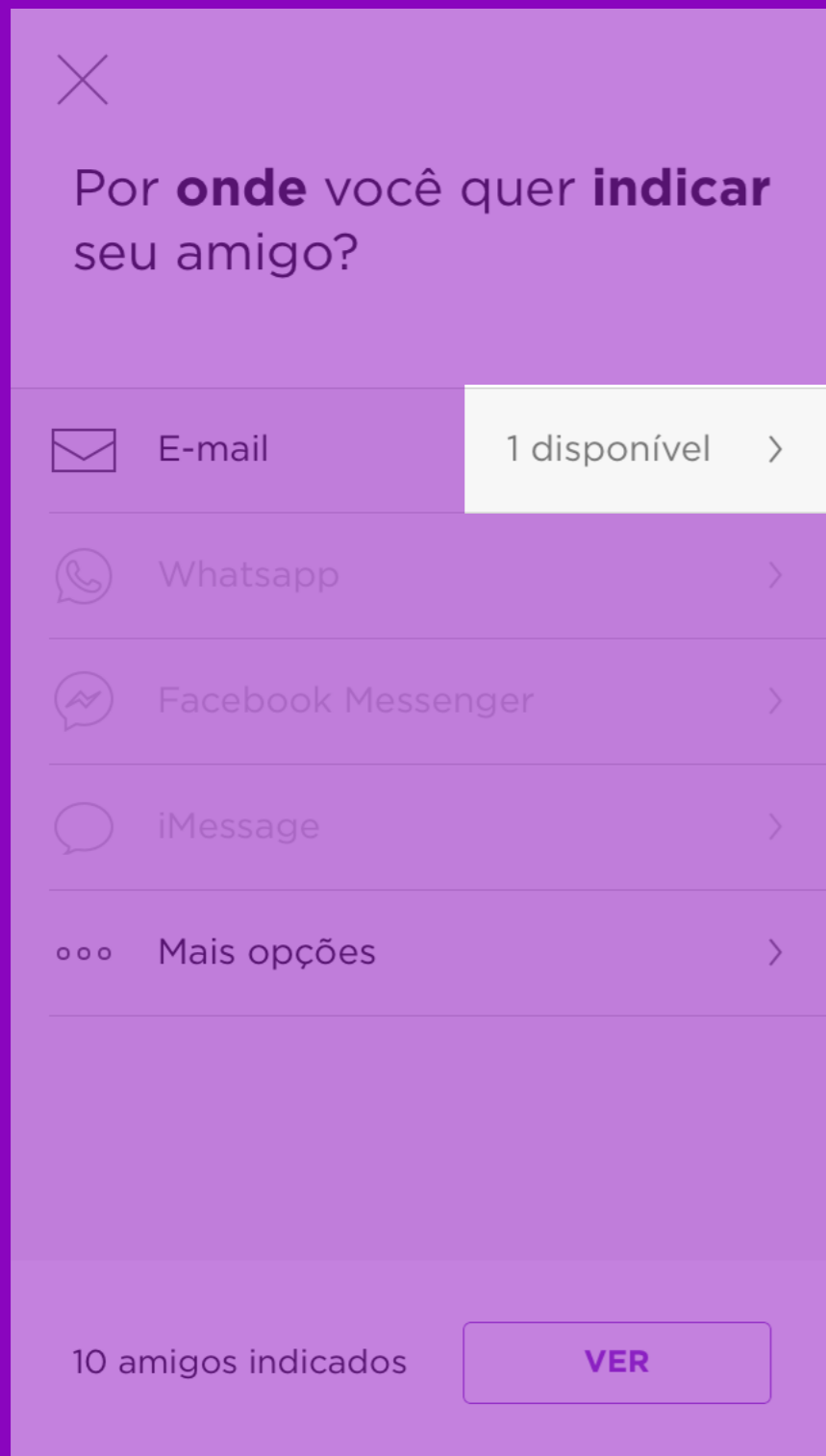


```
struct MGMChannelData {  
    let type: MGMChannel  
    let count: Int?  
    let canOpen: Bool  
}
```

Testes unitários

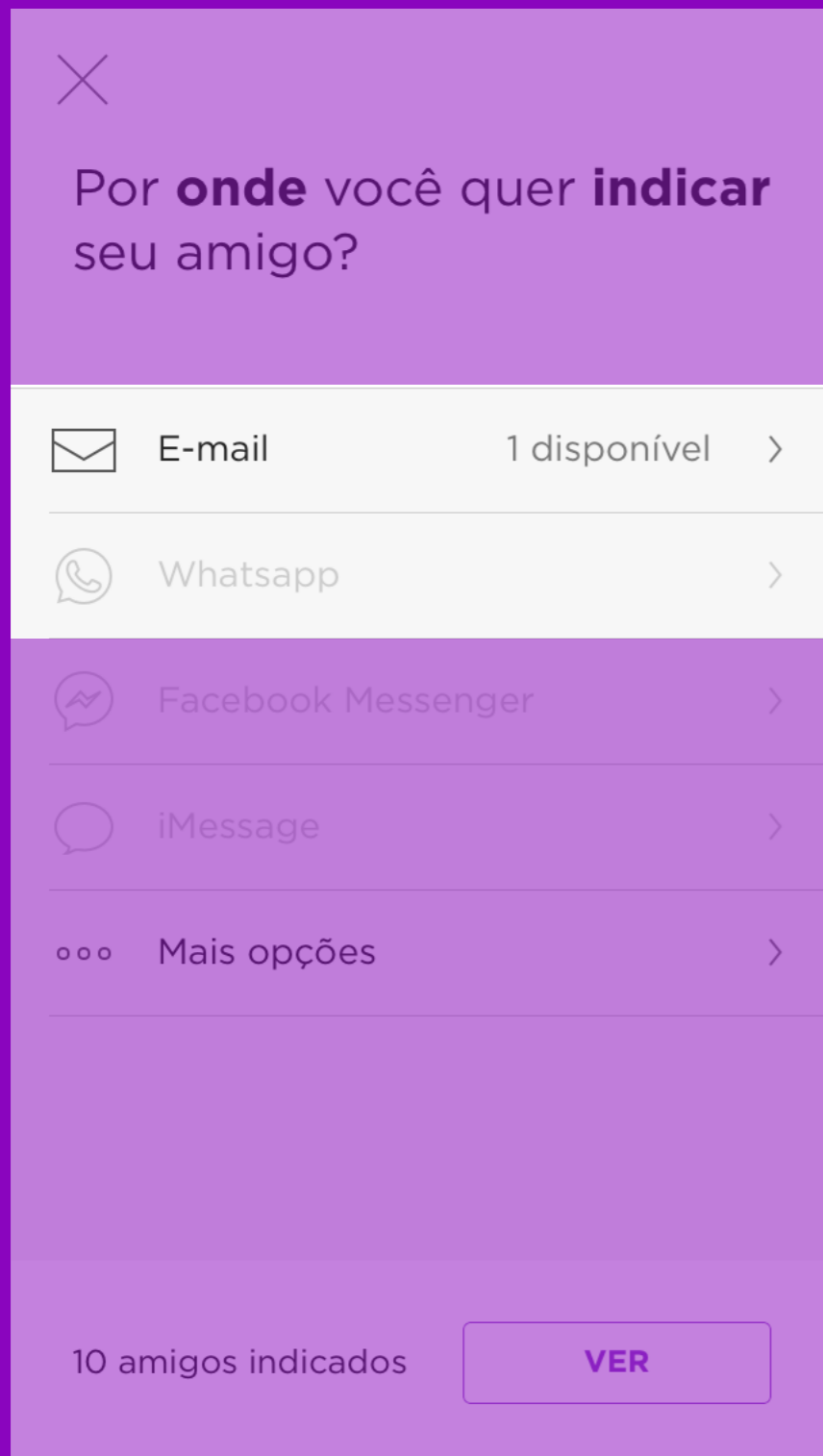
Integração

Unitários



```
struct MGMChannelData {  
    let type: MGMChannel  
    let count: Int?  
    let canOpen: Bool  
}
```

Testes unitários



```
struct MGMChannelData {  
    let type: MGMChannel  
    let count: Int?  
    let canOpen: Bool  
}
```

Testes unitários

Exemplo: Funções puras



```
struct MGMChannelData {  
  let type: MGMChannel  
  let count: Int?  
  let canOpen: Bool  
}  
  
class MGMInteractor {  
  ...  
  func createChannelsData(urlHandler: URLHandler,  
                           invitations: Int,  
                           channels: [Customer.Channel]) -> [MGMChannelData]  
}
```

Testes unitários

Exemplo: Funções puras



```
struct MGMChannelData {
  let type: MGMChannel
  let count: Int?
  let canOpen: Bool
}

class MGMInteractor {
  ...
  func createChannelsData(urlHandler: URLHandler,
                          invitations: Int,
                          channels: [Customer.Channel]) -> [MGMChannelData]
}

func testChannelDataCreationEmailNoInvites() {

}
```


Testes unitários

Exemplo: Funções puras



```
struct MGMChannelData {
  let type: MGMChannel
  let count: Int?
  let canOpen: Bool
}

class MGMInteractor {
  ...
  func createChannelsData(urlHandler: URLHandler,
                          invitations: Int,
                          channels: [Customer.Channel]) -> [MGMChannelData]
}

func testChannelDataCreationEmailPositiveNumberOfInvitesCanOpen() {

}
```


Testes unitários

Exemplo: Efeitos colaterais

Integração

Unitários

```
func testInviteMGM() {
  let mgmManager = StubMGManager()
  let stubViewController = StubViewController()
  let controller = Controller(accountStatus: .just(.active),
                              mgmManager: mgmManager,
                              viewController: stubViewController)

  //Given a simulated tap
  stubViewController.confirmSubject.onNext(())

  //Expect side-effects
  expect(mgmManager.inviteInvocationCount) == 1
}
```

Testes unitários

Exemplo: Efeitos colaterais



```
func testInviteMGM() {
  let mgmManager = StubMGManager()
  let stubViewController = StubViewController()
  let controller = Controller(accountStatus: .just(.active),
                              mgmManager: mgmManager,
                              viewController: stubViewController)

  //Given a simulated tap
  stubViewController.confirmSubject.onNext(())

  //Expect side-effects
  expect(mgmManager.inviteInvocationCount) == 1
}
```

Testes unitários

Exemplo: Efeitos colaterais

Integração

Unitários

```
func testInviteMGM() {
  let mgmManager = StubMGManager()
  let stubViewController = StubViewController()
  let controller = Controller(accountStatus: .just(.active),
                              mgmManager: mgmManager,
                              viewController: stubViewController)

  //Given a simulated tap
  stubViewController.confirmSubject.onNext(())

  //Expect side-effects
  expect(mgmManager.inviteInvocationCount) == 1
}
```


Testes unitários de view



- Permite validar o layout em várias condições de uso
 - Uso de screenshots
 - Facilita a verificação com o Design
- Evita modificações não intencionais no layout
- Permite validar a acessibilidade das telas

Testes unitários de view

Exercitar todos os **tamanhos** da tela ou componente



4S



5/5S/5C/SE



6/6S/7/8



6P/7P/8P



X

* Telas não necessariamente verdadeiras

Testes unitários de view



Exercitar todos os **tamanhos de fonte** na tela ou componente

Digite o **código** do seu convite

some-code

CONFIRMAR

CategoryXS

Digite o **código** do seu convite

some-code

CONFIRMAR

CategoryXXXL

Digite o **código** do seu convite

some-code

CONFIRMAR

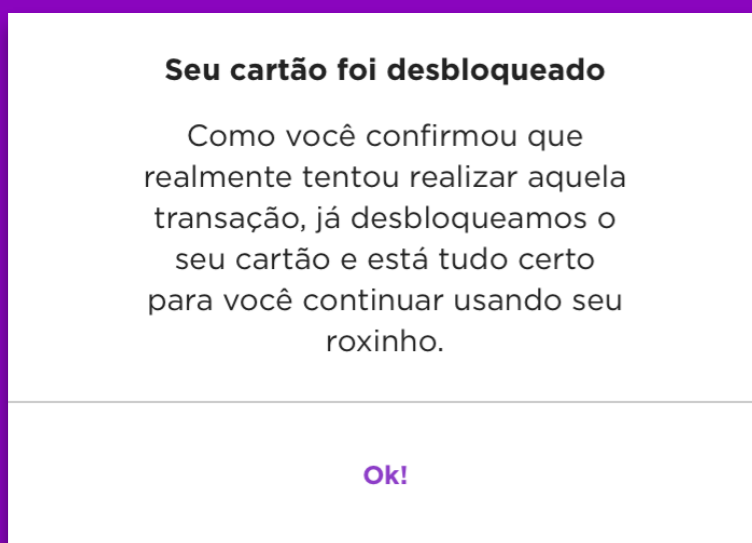
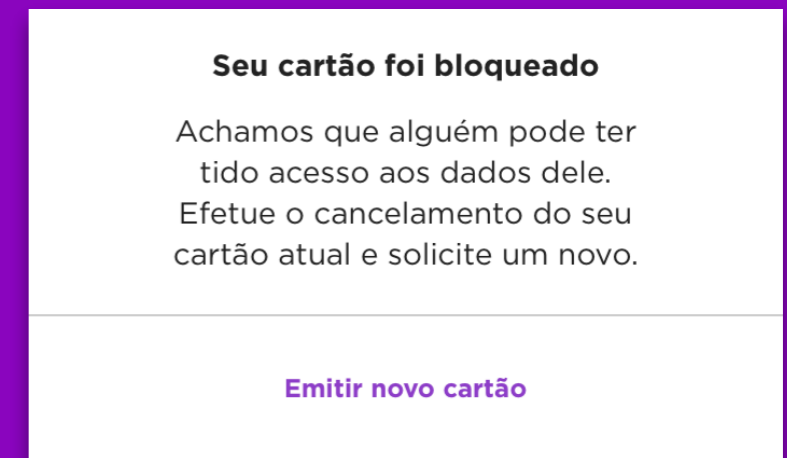
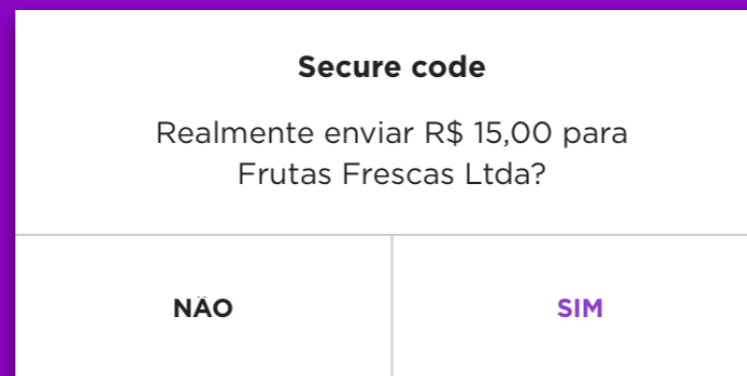
CategoryAccessibilityXXXL

* Telas não necessariamente verdadeiras

Testes unitários de view



Exercitar todos os **estados** da tela ou componente



* Telas não necessariamente verdadeiras

Testes unitários de view

Integração

Unitários

Exemplo

```
func testLayout() throws {
    let permutations = [
        ("popup", PopupViewModel(popup: somePopup)),
        ("one-action", PopupViewModel(popup: oneActionPopup)),
        ("no-icon", PopupViewModel(popup: noIconPopup)),
        ("recognized-timeout", .recognizedTimeout(url: .empty())),
        ("unrecognized-block", .unrecognizedBlock(url: .empty()))
    ]

    try permutations.forEach { identifier, viewModel in
        let viewController = PopupViewController()
        viewController.bind(viewModel)

        try assertView(viewController.view,
            width: 350,
            identifier: identifier,
            screenshotService: service)
    }
}
```


Testes unitários de view

Integração

Unitários

Exemplo

```
func testLayout() throws {
    let permutations = [
        ("popup", PopupViewModel(popup: somePopup)),
        ("one-action", PopupViewModel(popup: oneActionPopup)),
        ("no-icon", PopupViewModel(popup: noIconPopup)),
        ("recognized-timeout", .recognizedTimeout(url: .empty())),
        ("unrecognized-block", .unrecognizedBlock(url: .empty()))
    ]

    try permutations.forEach { identifier, viewModel in
        let viewController = PopupViewController()
        viewController.bind(viewModel)

        try assertView(viewController.view,
            width: 350,
            identifier: identifier,
            screenshotService: service)
    }
}
```

Testes unitários de view

Integração

Unitários

Exemplo

```
func testLayout() throws {
    let permutations = [
        ("popup", PopupViewModel(popup: somePopup)),
        ("one-action", PopupViewModel(popup: oneActionPopup)),
        ("no-icon", PopupViewModel(popup: noIconPopup)),
        ("recognized-timeout", .recognizedTimeout(url: .empty())),
        ("unrecognized-block", .unrecognizedBlock(url: .empty()))
    ]

    try permutations.forEach { identifier, viewModel in
        let viewController = PopupViewController()
        viewController.bind(viewModel)

        try assertView(viewController.view,
            width: 350,
            identifier: identifier,
            screenshotService: service)
    }
}
```

Testes unitários de view

Integração

Unitários

Exemplo

```
func testLayout() throws {
    let permutations = [
        ("popup", PopupViewModel(popup: somePopup)),
        ("one-action", PopupViewModel(popup: oneActionPopup)),
        ("no-icon", PopupViewModel(popup: noIconPopup)),
        ("recognized-timeout", .recognizedTimeout(url: .empty())),
        ("unrecognized-block", .unrecognizedBlock(url: .empty()))
    ]

    try permutations.forEach { identifier, viewModel in
        let viewController = PopupViewController()
        viewController.bind(viewModel)

        try assertView(viewController.view,
            width: 350,
            identifier: identifier,
            screenshotService: service)
    }
}
```


Testes de integração

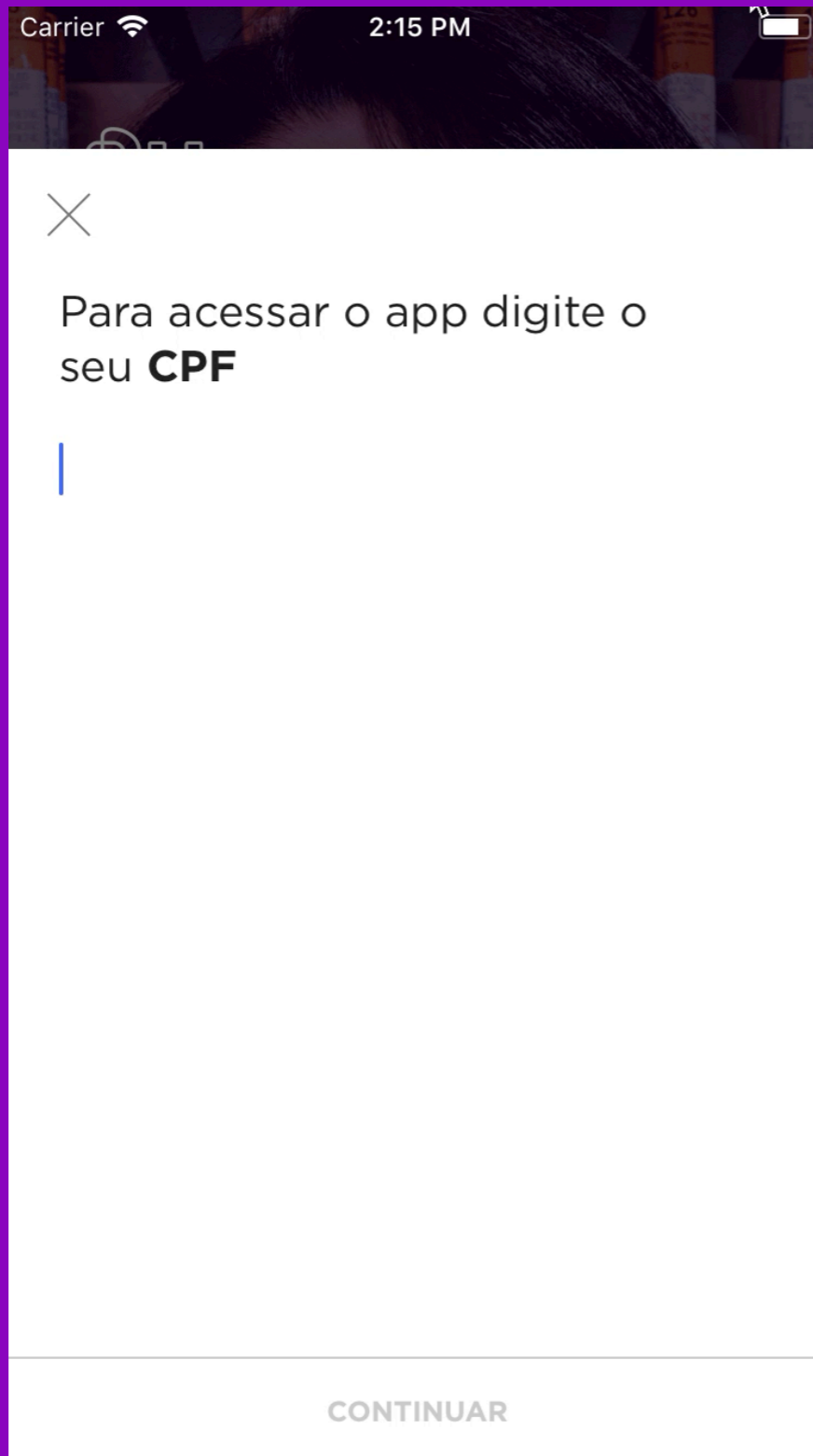


- Foco no funcionamento dos componentes *em conjunto*
- Fluxos reais do app com poucas modificações
- Requests *mockados* internamente no teste

Testes de integração

Integração

Unitários



Testes de integração

Integração

Unitários

Exemplo

```
func login() {
    tester().tapView(withAccessibilityLabel: "LOGIN")

    tester().waitForSoftwareKeyboard()
    tester().enterText(intoCurrentFirstResponder: validCPF)
    tester().tapView(withAccessibilityLabel: "CONTINUAR")

    tester().enterText(intoCurrentFirstResponder: validPassword)
    tester().tapView(withAccessibilityLabel: "ENTRAR")
}

func testReorderFlow() throws {
    tester().swipeView(withAccessibilityIdentifier: "global-actions",
                      inDirection: .left)

    tester().moveRow(at: IndexPath(row: 0, section: 0),
                    to: IndexPath(row: 1, section: 0),
                    in: tester().firstTableView())

    tester().waitForAnimationsToFinish()

    tester().tapView(withAccessibilityLabel: "Voltar")
    tester().swipeView(withAccessibilityIdentifier: "global-actions",
                      inDirection: .right)
    tester().waitForAnimationsToFinish()
}
```

Testes de integração

Integração

Unitários

Exemplo

```
func login() {
    tester().tapView(withAccessibilityLabel: "LOGIN")

    tester().waitForSoftwareKeyboard()
    tester().enterText(intoCurrentFirstResponder: validCPF)
    tester().tapView(withAccessibilityLabel: "CONTINUAR")

    tester().enterText(intoCurrentFirstResponder: validPassword)
    tester().tapView(withAccessibilityLabel: "ENTRAR")
}

func testReorderFlow() throws {
    tester().swipeView(withAccessibilityIdentifier: "global-actions",
                      inDirection: .left)

    tester().moveRow(at: IndexPath(row: 0, section: 0),
                    to: IndexPath(row: 1, section: 0),
                    in: tester().firstTableView())

    tester().waitForAnimationsToFinish()

    tester().tapView(withAccessibilityLabel: "Voltar")
    tester().swipeView(withAccessibilityIdentifier: "global-actions",
                      inDirection: .right)
    tester().waitForAnimationsToFinish()
}
```

Perguntas?

Victor Maraccini

@vmaraccini | victor.maraccini@nubank.com.br

Francesco Perrotti-Garcia

@fpg1503 | francesco.garcia@nubank.com.br

Obrigado!



sou.nu/jobs-at-nubank

Victor Maraccini

@vmaraccini | victor.maraccini@nubank.com.br

Francesco Perrotti-Garcia

@fpg1503 | francesco.garcia@nubank.com.br

niy bank