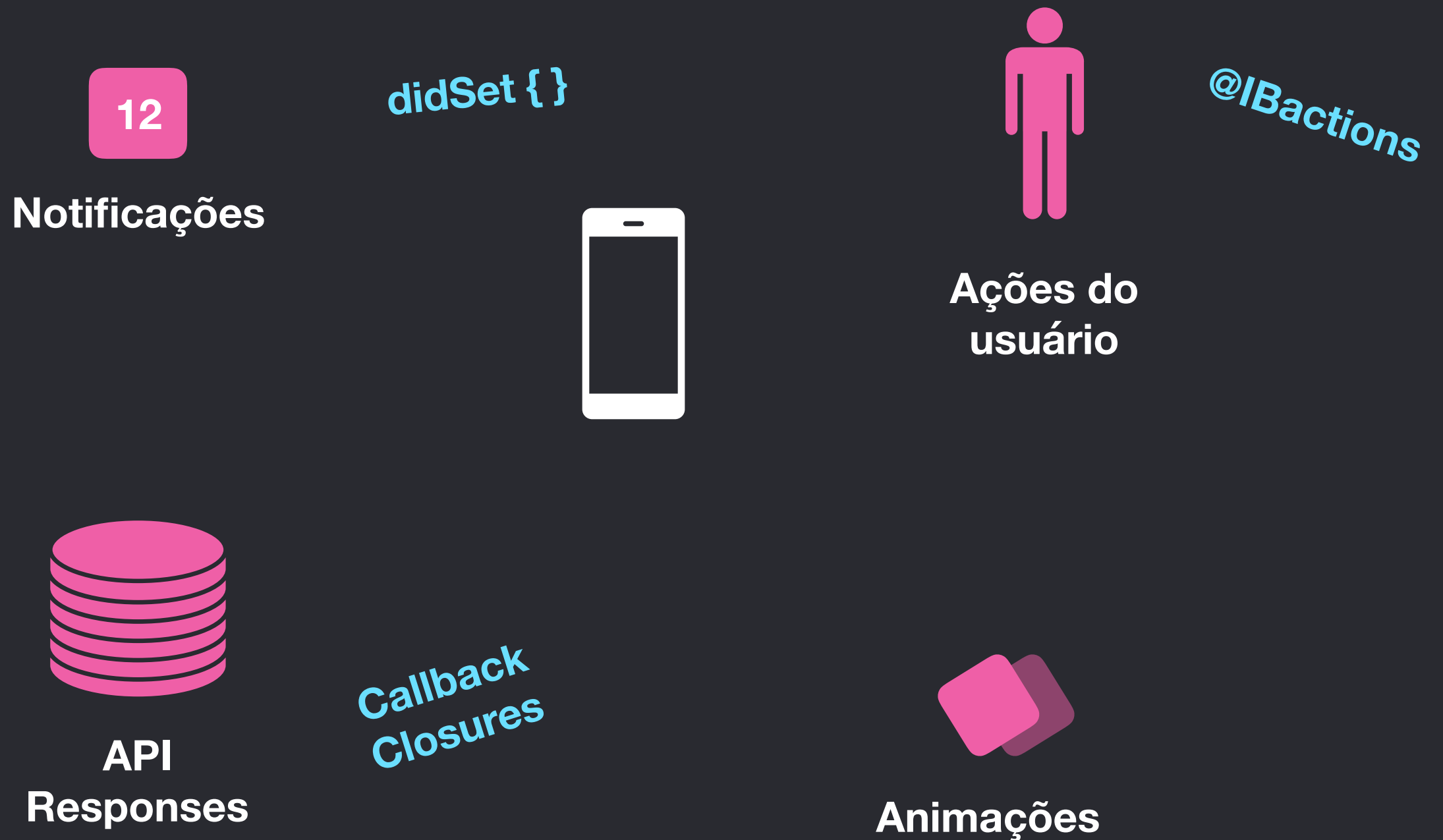


Combine

Laura Corssac

Programação assíncrona



Combine

- Framework declarativo e reativo
- WWDC 2019
- Processar valores assíncronos
- Alternativa ao RxSwift

Publisher

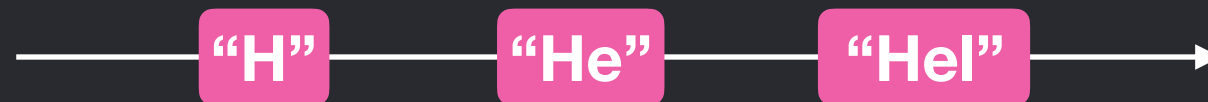
RX: Observable



Publisher

RX: Observable

```
@Published var typedText: String = ""
```



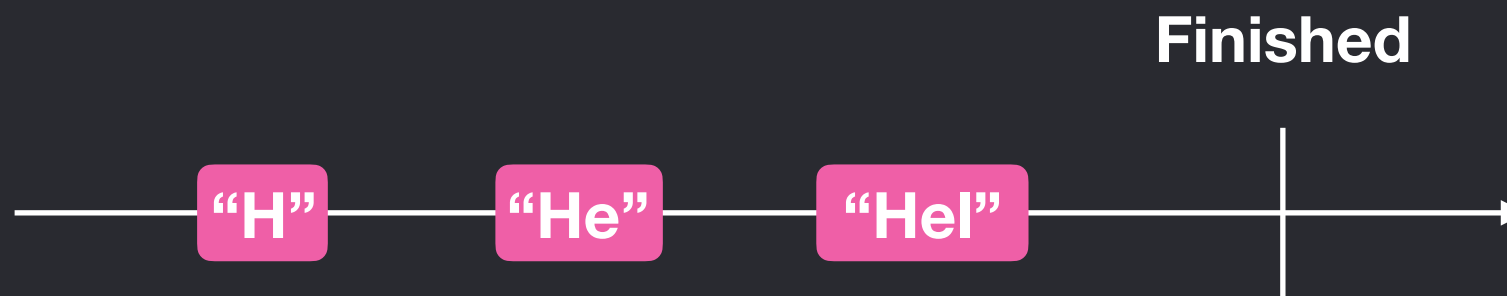
Publisher

RX: Observable

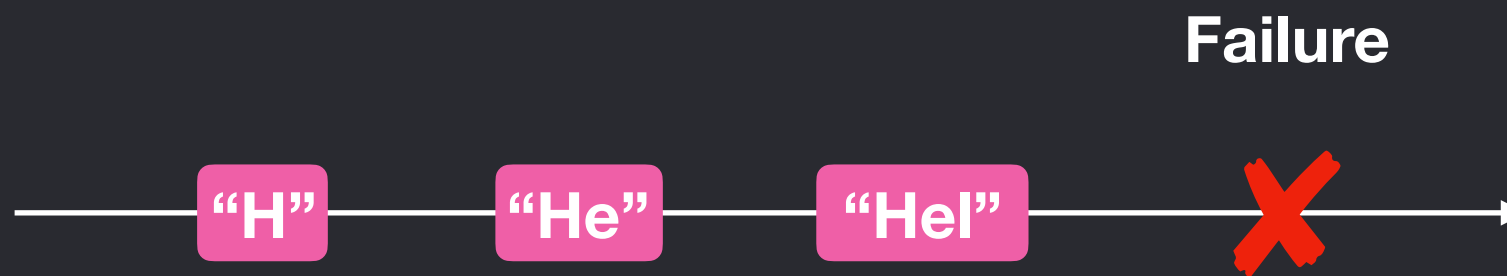
```
NotificationCenter.default.publisher(for: Notification.Name("hello"))
```



Término



Término



Término

```
public enum Completion<Failure> where Failure : Error {  
    case finished  
  
    case failure(Failure)  
}
```

Publisher

- **associatedtype** Output
- **associatedtype** Failure

Subscriber

- **associatedtype** **Input**
- **associatedtype** **Failure**

Demanda

- `.none`
- `.max(n)`
- `.unlimited`

Subscribers

```
final class IntSubscriber: Subscriber {
    typealias Input = Int
    typealias Failure = Never

    func receive(subscription: Subscription) {
        subscription.request(.max(3))
    }

    func receive(_ input: Int) -> Subscribers.Demand {
        print("Received value", input)
        return .none
    }

    func receive(completion: Subscribers.Completion<Never>) {
        print("Received completion", completion)
    }
}

let subscriber = IntSubscriber()

publisher.subscribe(subscriber)
```

Subscribers

```
_ = publisher.sink(receiveCompletion: { completion in
    switch completion {
    case .finished:
        print("publisher finished")
    case let .failure(error):
        print("publisher finished with error", error)
    }
}) { value in
    print("received value", value)
}
```

Subscribers

ViewModel:

```
@Published var inputIsValid = false
```

ViewController:

```
viewModel.$inputIsValid  
    .assign(to: \.isEnabled, on: continueButton)  
    .store(in: &subscriptions)
```

Operators

- Métodos do Publisher
- Mudam o tipo do Erro e/ou Output
- Combinados

Operators

Arrays

```
let newArray =  
  [1,2,3,4]  
  .map { $0 + 1 }
```

Publishers

```
let newPublisher =  
  publisher  
  .map { $0 + 1 }
```

Operators

Publishers

```
enum CustomError: Error {  
    case error10  
    case generic  
}  
  
let newPublisher = publisher.mapError {  
    $0.code == 10 ? CustomError.error10 : CustomError.generic  
}
```

Conclusão

	RxSwift	Combine
BackPressure	✗	✓
Retrocompatibilidade	✓	✗
Erros genéricos	✗	✓
Velocidade	😓	😊
Manutenção	Comunidade	Apple

Referências

- try! Swift NYC 2019 - Getting Started With combine
 - <https://www.youtube.com/watch?v=fejmsuxolvA&t=317s>
- Ray Werderlich
- Sessions: Combine in Practice e Introducing Combine
- RxSwift to Apple's Combine “Cheat Sheet”
 - <https://medium.com/gett-engineering/rxswift-to-apples-combine-cheat-sheet-e9ce32b14c5b>
- Will Combine Kill RxSwift?
 - <https://medium.com/flawless-app-stories/will-combine-kill-rxswift-64780a150d89>



@lauracorssac



Laura Corssac

