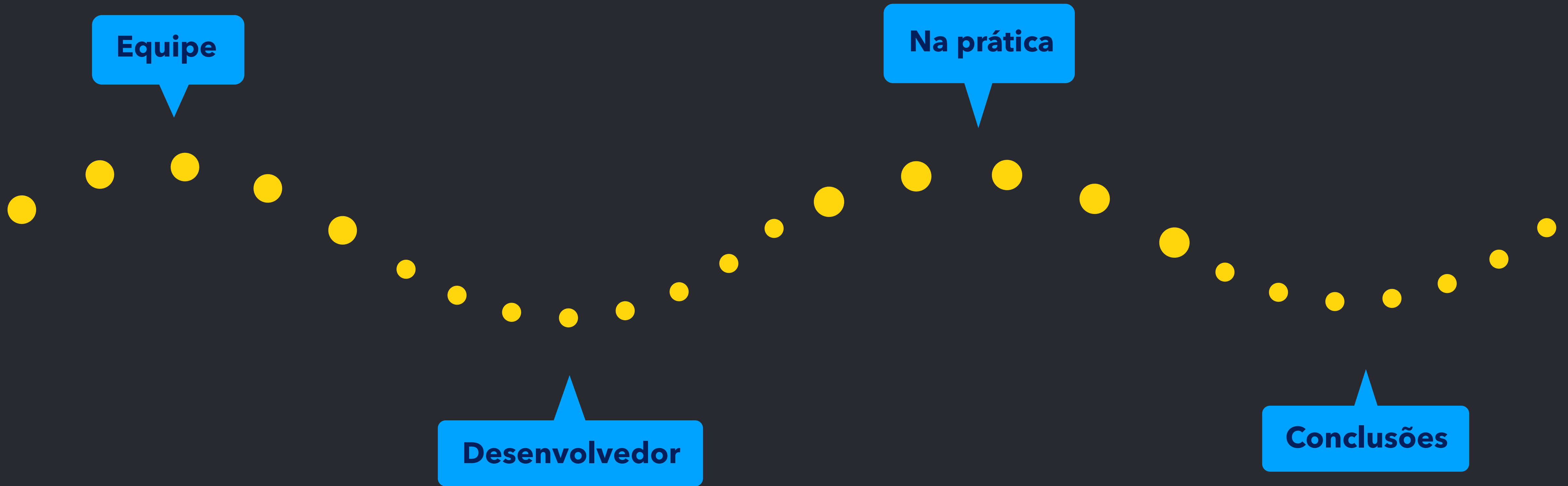


# **ADICIONANDO TESTES A APLICATIVOS EM PRODUÇÃO**

# motivação

---



**Equipe**

**Na prática**

**Desenvolvedor**

**Conclusões**

**equipe**

---



**acordo**



**nivelamento**

Time





Time



**desenvolvedor**

---

*DEFINITION OF DONE*

**GARANTIR QUALIDADE**

**TESTES NÃO SÃO FEATURES**

**desenvolvimento**

---



**legadíssimo**

**sem testes**



# Cenário escolhido

---



# Nova camada

---

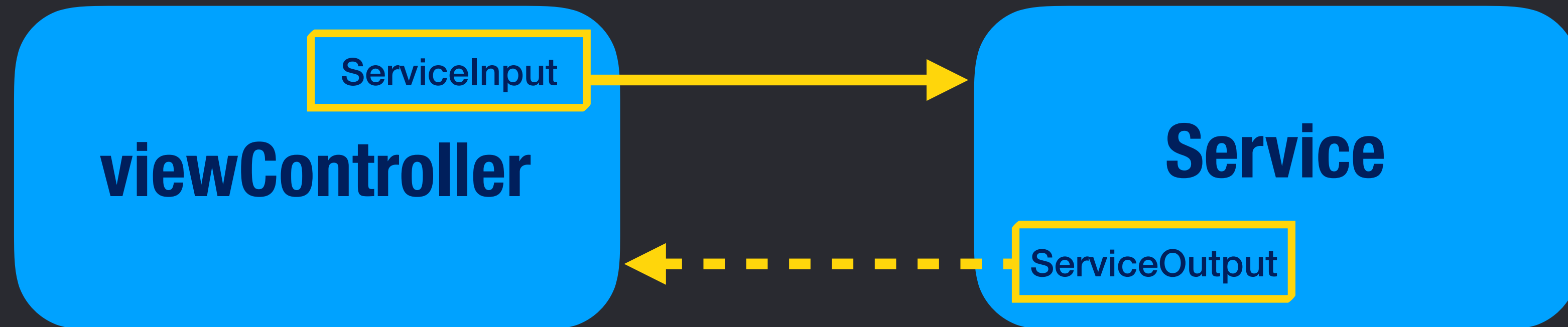
```
var service: ServiceInput?
```

```
service?.requestData()
```

```
weak var output: ServiceOutput?
```

```
output?.requestSucceeded()
```

```
output?.requestFailed()
```



# API atual

---

```
class Service {  
    let api: API  
    init (api: API) { ... }  
}  
  
extension Service: ServiceInput {  
    weak var output: ServiceOutput?  
    func requestData() {  
        api.request( parâmetros ) { ... }  
    }  
}
```

# API via interface

---

```
protocol APIProvider: AnyObject {  
    func request(  
        url: String,  
        httpMethod: HttpMethod,  
        completion: @escaping (Result) -> Void  
    )  
}
```



# Interfaces

---

```
protocol ServiceInput: AnyObject {  
    var output: ServiceOutput? {get set}  
  
    func requestData()  
}
```

```
protocol ServiceOutput: AnyObject {  
    func requestSucceeded(content: String)  
    func requestFailed(error: String)  
}
```

# service

---

```
class Service {  
    let api: APIProvider  
    init (api: APIProvider) { ... }  
}  
  
extension Service: ServiceInput {  
    weak var output: ServiceOutput?  
    func requestData() {  
        api.request( parâmetros ) { ... }  
    }  
}
```

# VC - antes/depois

---

```
class MyViewController: UIViewController {  
  
    /*  
        LAYOUT  
        ...  
        REGRAS DE APRESENTAÇÃO  
        ...  
        SERVIÇO  
    */  
}
```

# o que testar?

---

- endpoint
- parâmetros
- método http
- delegar sucesso
- delegar erro

# API Spy

---

```
class APIProviderSpy: APIProvider {  
  
    var urlPassed: String?  
    var httpMethodPassed: HttpMethod?  
    var requestDataCalled: Bool?  
  
    var status: Result?  
  
    func request(  
        url: String,  
        httpMethod: HttpMethod,  
        completion: @escaping (Result) -> Void) {...}  
  
}
```



# testando service

---

```
class ServiceTests: XCTestCase {  
  
    var service: Service!  
    var outputSpy: ServiceOutputSpy!  
    var apiSpy: APIProviderSpy!  
  
    override func setUp() {  
        outputSpy = ServiceOutputSpy()  
        apiSpy = APIProviderSpy()  
        service = Service(api: apiSpy)  
        service.output = outputSpy  
    }  
}
```

# teste de parâmetros

---

```
func test_When_requestData_Called() {  
    service.requestData()  
  
    XCTAssert(apiSpy.urlPassed == "www.myendpoint.com")  
    XCTAssert(apiSpy.httpMethodPassed == .get)  
    XCTAssert(apiSpy.requestDataCalled == true)  
}
```

```
override func setUp() {  
    outputSpy = ServiceOutputSpy()  
    apiSpy = APIProviderSpy()  
    service = Service(api: apiSpy)  
    service.output = outputSpy  
}
```

# observando retorno

---

```
class ServiceOutputSpy: ServiceOutput {  
  
    var dataPassed: String?  
    var requestSucceededCalled: Bool?  
    var errorPassed: String?  
    var requestFailedCalled: Bool?  
  
    func requestSucceeded(content: String) {  
        dataPassed = content  
        requestSucceededCalled = true  
    }  
  
    func requestFailed(error: String) {  
        errorPassed = error  
        requestFailedCalled = true  
    }  
}
```

# teste de retorno

---

```
func test_When_requestData_Failed() {
    apiSpy.status = .failed("Erro inesperado")
    service.requestData()

    XCTAssert(outputSpy.errorPassed == "Erro inesperado")
    XCTAssert(outputSpy.requestFailedCalled == true)
}
```

```
override func setUp() {
    outputSpy = ServiceOutputSpy()
    apiSpy = APIProviderSpy()
    service = Service(api: apiSpy)
    service.output = outputSpy
}
```

**conclusões**

---






## Victor Panitz Magalhães

victorpanitz

iOS Engineer

 Agibank

 Porto Alegre

 [victorpanitz@gmail.com](mailto:victorpanitz@gmail.com)



[/victorpmagalhaes](#)



[/victormagalhaes](#)



[/victorpanitz](#)

# OBRIGADO