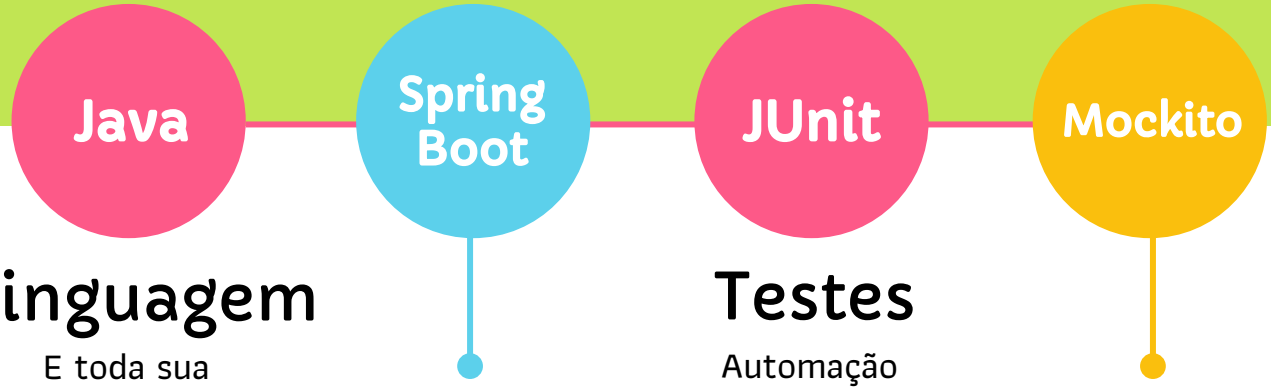
Escolha seu sabor de
Testes Unitários com

Kotest e MockK

Por que estamos aqui hoje?

“Nada é permanente, exceto a mudança”
Heráclito

NOSSA STACK



Linguagem

E toda sua plataforma acoplada

Framework

JEE, inversão de controle, injeção de dependência

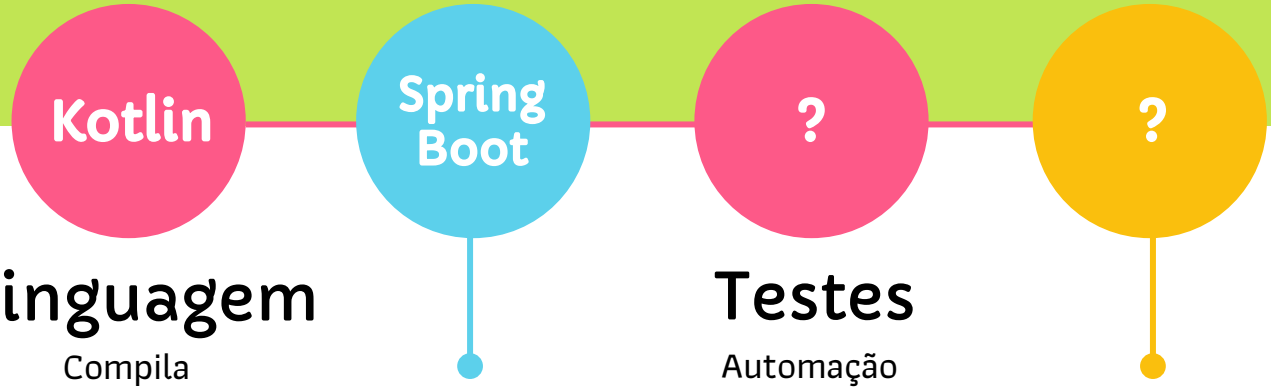
Testes

Automação

Mocks

Mocking framework

NOSSA NOVA STACK?



Linguagem

Compila para a JVM

Framework

JEE, inversão de controle, injeção de dependência

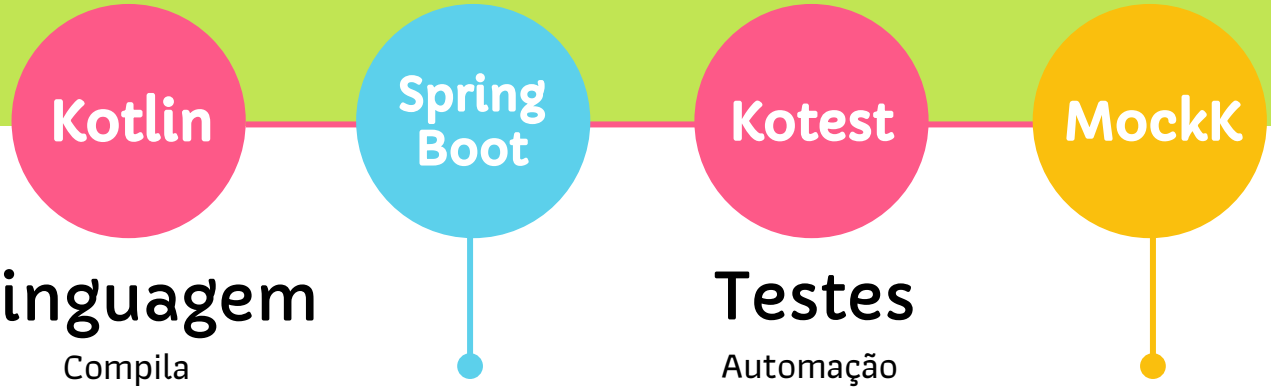
Testes

Automação

Mocks

Mocking framework

NOSSA NOVA STACK!



Linguagem

Compila para a JVM

Framework

JEE, inversão de controle, injeção de dependência

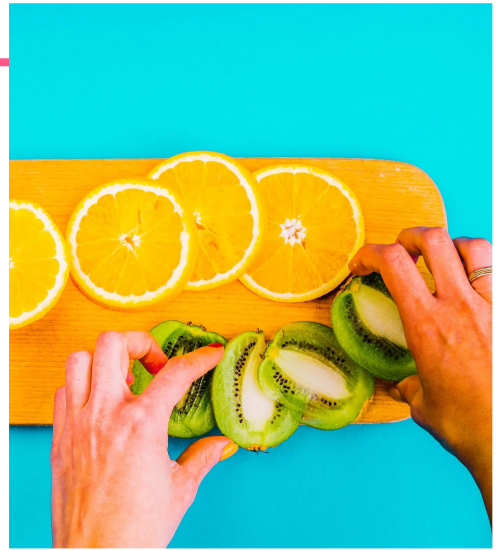
Testes

Automação

Mocks

Mocking framework

Spring Boot



Permite criar de maneira fácil aplicações stand-alone.

O servidor vai embutido na aplicação.

Mínimo de configuração necessária para funcionar.

Project

- (●) Maven Project
- ( ) Gradle Project

Language

- ( ) Java
- (●) Kotlin
- ( ) Groovy

Spring Boot

- ( ) 2.5.1 (SNAPSHOT)
- (●) 2.5.0
- ( ) 2.4.7 (SNAPSHOT)
- ( ) 2.4.6
- ( ) 2.3.12 (SNAPSHOT)
- ( ) 2.3.11

Project Metadata

Group	com.example
Artifact	demo
Name	demo
Description	Demo project for Spring Boot

Dependencies

ADD DEPENDENCIES... CTR

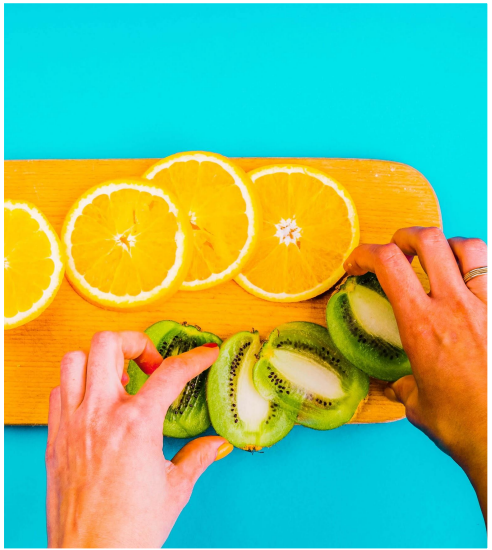
Spring Web WEB

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Testes no Spring

Pela dependência
spring-boot-starter-test

Algumas Annotations usadas:
@SpringBootTest
@WebMvcTest
@TestConfiguration
@MockBean

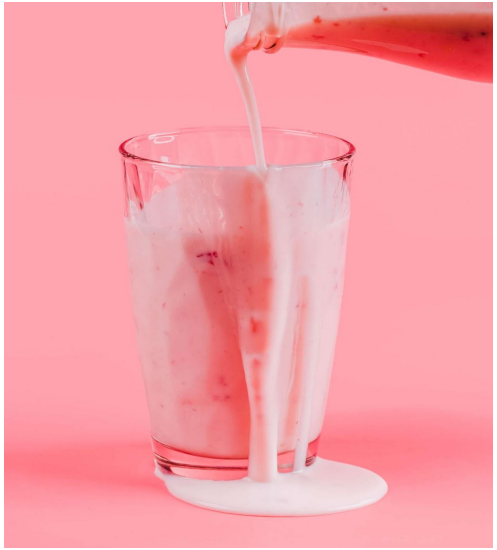


Kotlin

“Uma linguagem de programação moderna que deixa os desenvolvedores mais felizes”.

Linguagem moderna, concisa e **segura.**

Compatível e interoperável com o ecossistema Java.



MockK

Framework de criação de mocks para Kotlin. Tem um estilo muito próximo ao que o Mockito é para o Java.

“*Mock objects* são objetos que simulam o comportamento de objetos reais de forma controlada. São objetos “falsos” que simulam o comportamento de uma classe ou objeto “real” para que possamos focar o teste na unidade a ser testada”.

—Wikipedia

Como funciona?

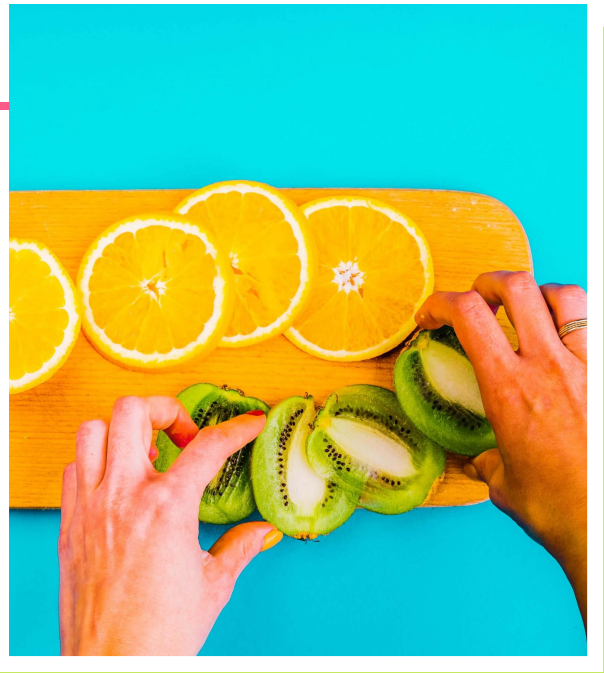
```
val car = mockk<Car>()

every { car.drive(Direction.NORTH) } returns Outcome.OK

car.drive(Direction.NORTH) // returns OK

verify { car.drive(Direction.NORTH) }
```

Integração com o Spring



Pelo projeto **SpringMockK**

Duas novas annotations:
@MockkBean
@SpykBean

Kotest

Framework de teste flexível e elegante, com biblioteca de asserções e biblioteca de teste de propriedade para Kotlin.

Kotest é modular

01

Testes

Permite que os testes sejam dispostos de uma forma fluida e executáveis na JVM ou Javascript em 10 estilos diferentes.

02

Asserções

Rica biblioteca com mais de 300 formas de asserções em um estilo DSL.

03

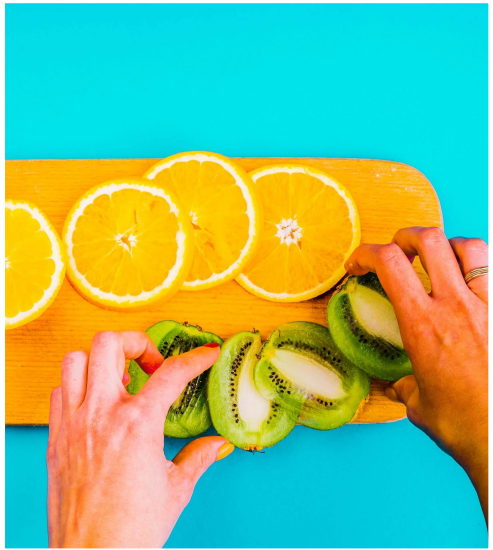
Propriedades

Recursos para a construção de testes baseados em valores e propriedades.

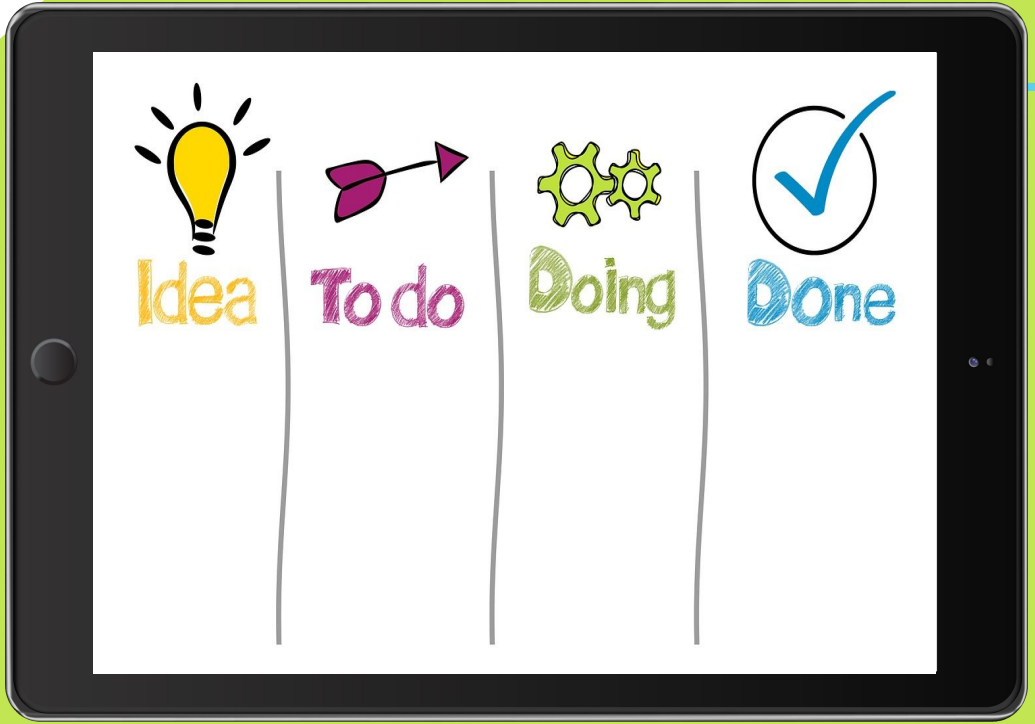
Integração com o Spring

Pela dependência
kotest-extensions-spring

```
override fun extensions() = listOf(SpringExtension)
```



Cenário de Teste



Backend de um aplicativo de quadro de tarefas estilo Kanban

Título	Arquivar Quadro Kanban
História de usuário	Eu como gestor de quadros Kanban quero poder arquivar quadros que não serão mais usados Para poder encerrar um quadro.
Critérios de aceitação	*Critério 1:* Dado que o usuário tenha o perfil de gestor de quadro E esteja na tela de listagem dos Quadros E que o quadro não tenha colunas Quando ele selecionar a opção arquivar quadro Então o quadro será arquivado e desaparece da listagem de quadros. *Critério 2:* Dado que o usuário tenha o perfil de gestor de quadro E esteja na tela de listagem dos Quadros E que o quadro tem colunas E estas não tenham tarefas ativas Quando ele selecionar a opção arquivar quadro Então o quadro será arquivado e desaparece da listagem de quadros. *Critério 3:* Dado que o usuário tenha o perfil de gestor de quadro E esteja na tela de listagem dos Quadros E que o quadro tem colunas E estas não tenham tarefas ou que as tarefas tenham sido finalizadas Quando ele selecionar a opção arquivar quadro Então aparecerá um alerta que o quadro não pode ser arquivado, pois ainda tem tarefas ativas.

Estilos (Sabores) de Testes

Fun

Should

Expect

String

Describe

Feature

Behavior

Annotation

Word

Free

Fun Spec

```
class BoardServiceFunSpecTest  : FunSpec() {
    init {
        test("Arquivar quadros sem colunas" ) {

        }

        test("Arquivar quadros com colunas e sem tarefas" ) {

        }

        test("Arquivar quadros sem colunas e com tarefas finalizadas" ) {

        }

        test("Arquivar quadros sem colunas e com tarefas ativas" ) {

        }
    }
}
```

Voltar

Should Spec

```
internal class BoardServiceShouldSpecTest : ShouldSpec() {
    init {
        should("Arquivar quadros sem colunas") {

        }

        should("Arquivar quadros com colunas e sem tarefas") {

        }

        should("Arquivar quadros sem colunas e com tarefas finalizadas")
{

        }

        should("Arquivar quadros sem colunas e com tarefas ativas") {

        }
    }
}
```

Voltar

Expect Spec

```
internal class BoardServiceExpectSpecTest : ExpectSpec() {
    init {
        expect ("Arquivar quadros sem colunas" ) {

        }
        expect("Arquivar quadros com colunas e sem tarefas" ) {

        }
        expect("Arquivar quadros sem colunas e com tarefas finalizadas" ) {

        }
        expect("Arquivar quadros sem colunas e com tarefas ativas" ) {

        }
    }
}
```

Voltar

String Spec

```
internal class BoardServiceStringSpecTest : StringSpec() {
    init {
        "Arquivar quadros sem colunas" {

        }
        "Arquivar quadros com colunas e sem tarefas" {

        }
        "Arquivar quadros sem colunas e com tarefas finalizadas" {

        }
        "Arquivar quadros sem colunas e com tarefas ativas" {

        }
    }
}
```

Voltar

Describe Spec

```
//JavaScript e RSpec style
internal class BoardServiceDescribeSpecTest  : DescribeSpec() {

    init {
        describe( "Arquivar quadros sem uso" ) {
            it("sem colunas") {

            }
            it(" com colunas e sem tarefas" ) {

            }
            it("sem colunas e com tarefas finalizadas" ) {

            }
            it("sem colunas e com tarefas ativas" ) {

            }
        }
    }
}
```

Voltar

Feature Spec

```
// Cucumber Style
internal class BoardServiceFeatureSpecTest  : FeatureSpec() {
   init {
       feature("Quadro") {
           scenario("Arquivar Quadro Kanban sem colunas" ) {

           }
           scenario("Arquivar Quadro Kanban com colunas sem tarefas ativas" ) {

           }
           scenario("Arquivar Quadro Kanban com colunas e tarefas ativas" ) {

           }
       }
   }
}
```

Voltar

Behavior Spec

```
internal class BoardServiceBehaviorSpecTest : BehaviorSpec() {
    init {
        given("que o usuário tenha o perfil de gestor de quadro") {
            and(" o quadro não possui colunas") {
                When( "ele selecionar a opção arquivar quadro") {
                    then(" o quadro será arquivado e desaparece da listagem de quadros.") {

                    }
                }
            }
            and(" o quadro possui colunas e estas não possuam tarefas ativas") {
                When( "ele selecionar a opção arquivar quadro") {
                    then(" o quadro será arquivado e desaparece da listagem de quadros.") {

                    }
                }
            }
            and(" o quadro possui colunas e estas possuam tarefas ativas") {
                When( "ele selecionar a opção arquivar quadro") {
                    then(" ocorrerá um alerta que o quadro não pode ser arquivado") {

                    }
                }
            }
        }
    }
}
```

Voltar

Describe Spec

```
//JavaScript e RSpec style
internal class BoardServiceDescribeSpecTest  : DescribeSpec() {

    init {
        describe( "Arquivar quadros sem uso" ) {
            it("sem colunas") {

            }
            it(" com colunas e sem tarefas" ) {

            }
            it("sem colunas e com tarefas finalizadas" ) {

            }
            it("sem colunas e com tarefas ativas" ) {

            }
        }
    }
}
```

Voltar

Annotation Spec

```
internal class BoardServiceAnnotationSpecTest : AnnotationSpec() {

    @BeforeEach
    fun setUp() {
    }

    @Test
    fun finishBoardWithNoColumns() {
    }

    @Test
    fun finishBoardWithColumnsAndNoTasks() {
    }

    @Test
    fun finishBoardWithColumnsAndNoActiveTasks() {
    }

    @Test
    fun finishBoardWithColumnsAndActiveTasks() {
    }
}
```

Voltar

Word Spec

```
internal class BoardServiceWordSpecTest : WordSpec() {
    init {
        "Arquivar quadros sem uso " When {
            "sem colunas" should {
                "com sucesso" {

                }
            }
            "com colunas e sem tarefas" should {
                "com sucesso" {

                }
            }
            "com colunas e com tarefas finalizadas" should {
                "com sucesso" {

                }
            }
            "com colunas e com tarefas ativas" should {
                "não deve ser permitido" {

                }
            }
        }
    }
}
```

Voltar

Free Spec

```
internal class BoardServiceFreeSpecTest : FreeSpec() {
    init {
        "Arquivar Quadros sem uso " - {
            "sem colunas" - {
                "com sucesso" {

                }
            }
            "com colunas e sem tarefas"- {
                "com sucesso" {

                }
            }
            "com colunas e com tarefas finalizadas"- {
                "com sucesso" {

                }
            }
            "com colunas e com tarefas ativas"- {
                "não deve ser permitido"{

                }
            }
        }
    }
}
```

Voltar

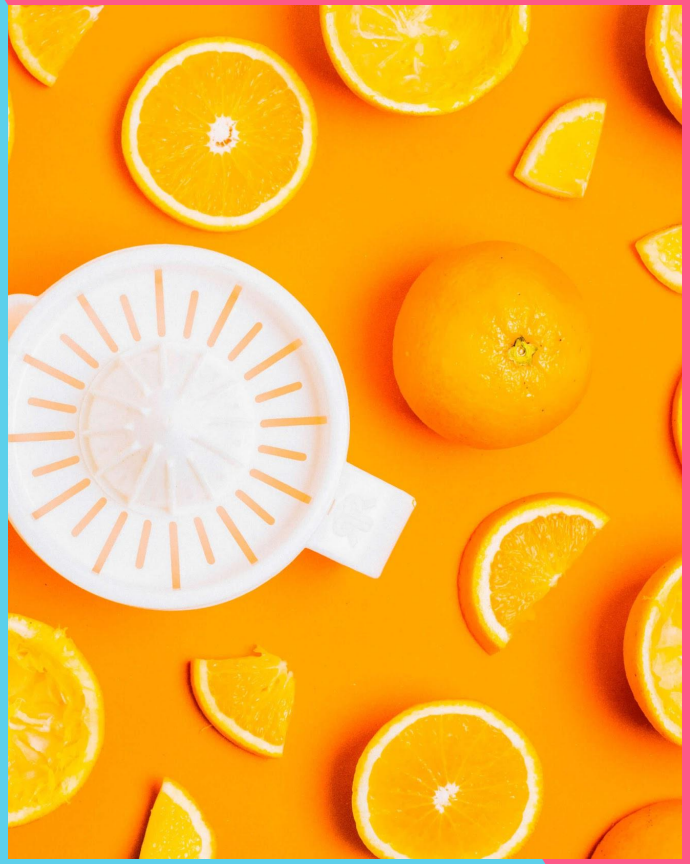
Demonstração

Links

Projeto exemplo:
https://github.com/alelima/ktboard

Kotest:
Kotest.io

MockK:
mockk.io



Obrigado!

alessandrolima@gmail.com

@alessandrolim

github.com/alelima