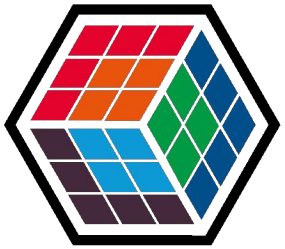


THE DEVELOPER'S CONFERENCE

Java - Construindo uma arquitetura de testes

Sandro L Giacomozzi
@sandrogiacom

Alisson Medeiros
@AlissonMedeiros

Por que?


Automação
Reduzir custos
Não existe
TCC
Mais controle
Preguiça
Necessidade
Comprar licenças
Inovação
Novidade
Meu é melhor
Produtividade
dependência externa
Repetição
hobby

THE DEVELOPER'S CONFERENCE

```java
public final class BooleanUtils {

    private static final Boolean TRUE = Boolean.TRUE;

    private static final Boolean FALSE = Boolean.FALSE;

    /**
     * This method is used to check if a boolean is true or false.
     *
     * @param isTrue The parameter to be checked if true.
     *
     * @return boolean This returns true if isTrue is true.
     */
    public static boolean checkIfTrue(boolean isTrue){
        if(isTrue == true){
            return TRUE;
        }else{
            return FALSE;
        }
    }
}
```

Nossa motivação

- Novo modelo de desenvolvimento e entrega de serviços
- Pouca ou nenhuma cultura de testes
- Cada equipe possui um padrão e ferramentas distintas
- Demora no feedback dos testes. Testes quebrados
- Confusão sobre teste de unidade e teste de integração
- Criar uma cultura colaborativa

Resumindo...

THE DEVELOPER'S CONFERENCE

Não tínhamos segurança para entregar software no modelo proposto com qualidade e velocidade esperada!

O que é - proposta

THE DEVELOPER'S CONFERENCE

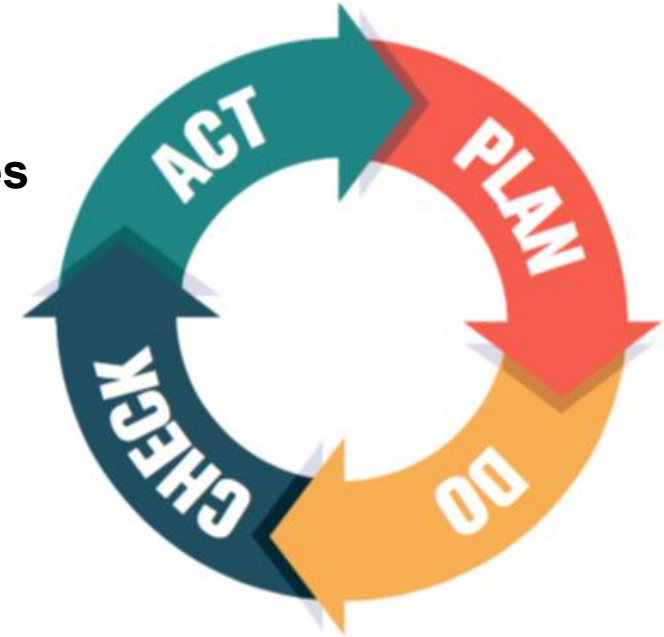
- **Conjunto de boas práticas**
  - Nomenclatura, estrutura de código
- **Ferramentas e frameworks**
  - JUnit, Mockito, AssertJ, Rest-assured, Testcontainers, Jacoco, PIT
- **Documentação**
  - Conceitos, como utilizar e exemplos

Primeira rodada

THE DEVELOPER'S CONFERENCE

Levante as necessidades e dificuldades da equipe

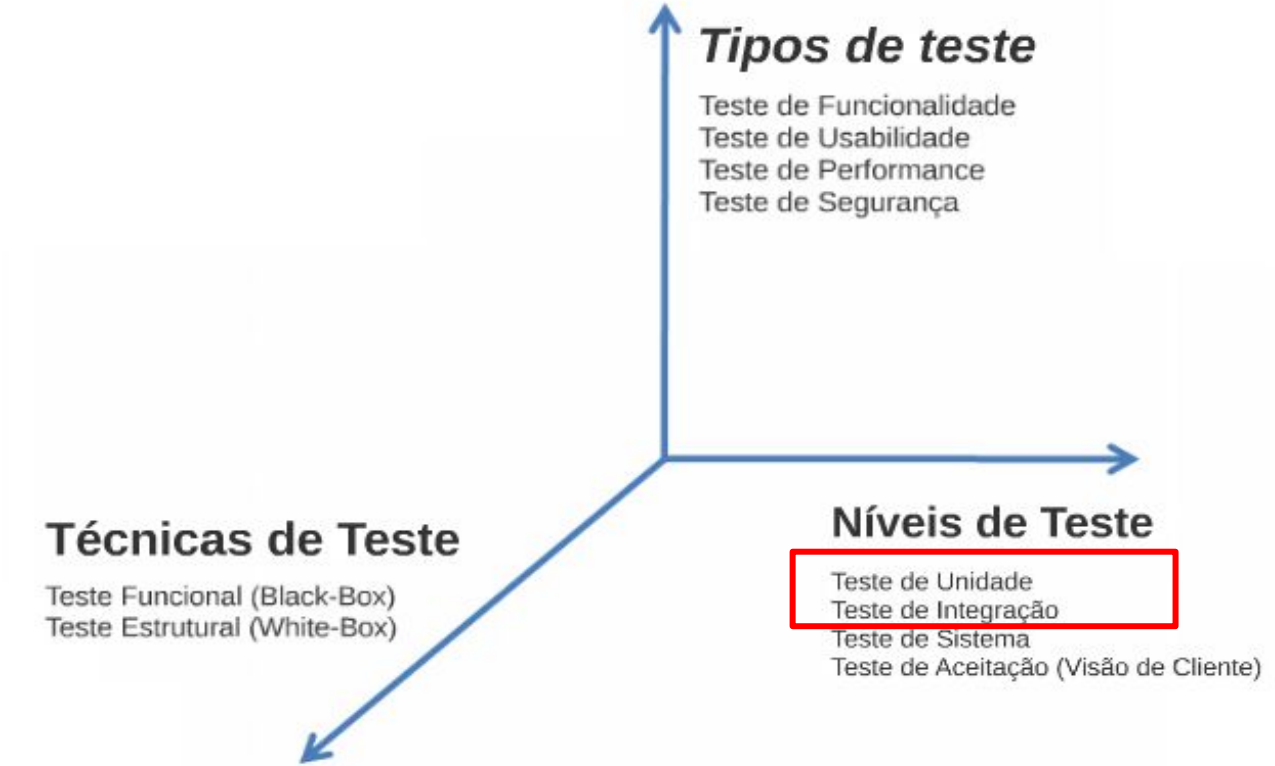
Faça uma implementação inicial (POC)



Valide com um projeto em desenvolvimento

Compartilhe os resultados e prepare novas funcionalidades

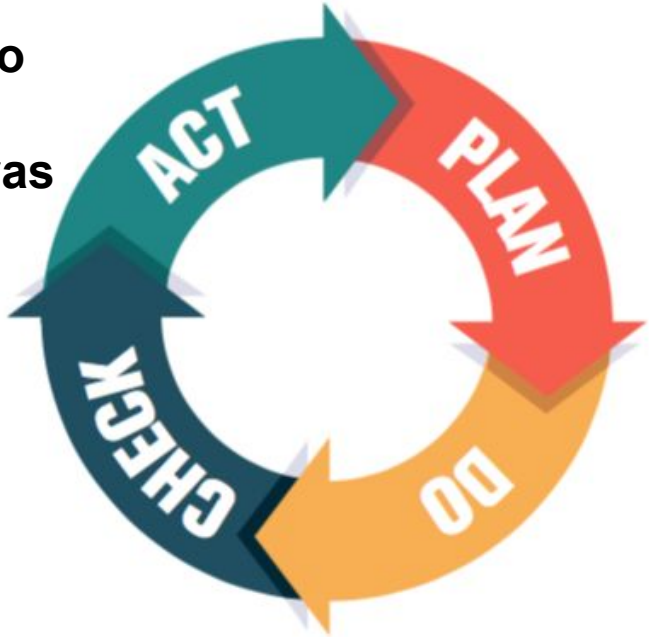
X de testes


Tipos de teste
Teste de Funcionalidade
Teste de Usabilidade
Teste de Performance
Teste de Segurança
Técnicas de Teste
Teste Funcional (Black-Box)
Teste Estrutural (White-Box)
Níveis de Teste
Teste de Unidade
Teste de Integração
Teste de Sistema
Teste de Aceitação (Visão de Cliente)

Exemplo

THE DEVELOPER'S CONFERENCE

Compartilhamento do aprendizado e preparação para novas funcionalidades.



Padrão para testes de unidade e integração.

Primeiros projetos utilizando o novo padrão

Criação do projeto com dependências Maven, exemplos e documentação.

Exemplo - Teste de Unidade

THE DEVELOPER'S CONFERENCE

```java
@Test
void whenMultiplyingThenCheckResult() {
    //Given
    CalcDemo calc = new CalcDemo();

    //When
    double result = calc.multiply(4, 5);

    //Then
    assertThat(result).isEqualTo(20);
}
```

JUnit + AssertJ

```java
@Mock
SendEmail sendEmail;

@InjectMocks
EmailService service;

@Test
void whenSendMailThenReturnTrue() {
    Mockito.when(sendEmail.send( email: "Teste")).thenReturn(true);

    boolean sent = service.sendEmail("Teste");

    Assertions.assertThat(sent).isTrue();
}
```

JUnit + AssertJ + Mockito

Exemplo - Teste de Integração

THE DEVELOPER'S CONFERENCE

```java
@RestController
public class TranslateController {

    private TranslatorService service;

    public TranslateController(TranslatorService service) {
        this.service = service;
    }

    @GetMapping({"/languages"})
    public Languages getLanguages() {
        Languages languages = service.getLanguages();
        return languages;
    }

}
```

```java
@Test
public void whenGetLanguagesThenVerifyTranslatorLanguages() {
    given() RequestSpecification
            .when() RequestSpecification
            .get( s: "/languages") Response
            .then() ValidatableResponse
            .statusCode(HttpStatus.OK.value()) ValidatableResponse
            .body( s: "languages", hasSize(104)) ValidatableResponse
            .body( s: "languages.language", hasItems("en", "pt", "es"));
}
```

Spring Boot Test + RestAssured

Exemplo - Teste de Integração com Mock

THE DEVELOPER'S CONFERENCE

```
@Test
public void whenGetLanguagesThenShowMockLanguages() throws Exception {
    GoogleLanguages langs = buildMockLanguages();

    MockWebIt.when(translate.getLanguages()).thenReturn(langs);

    given() RequestSpecification
            .when() RequestSpecification
            .get( s: "/languages") Response
            .then() ValidatableResponse
            .statusCode(HttpStatus.OK.value()) ValidatableResponse
            .body( s: "languages", hasSize(3)) ValidatableResponse
            .body( s: "languages.language", hasItems("en", "pt", "es"));
}
```

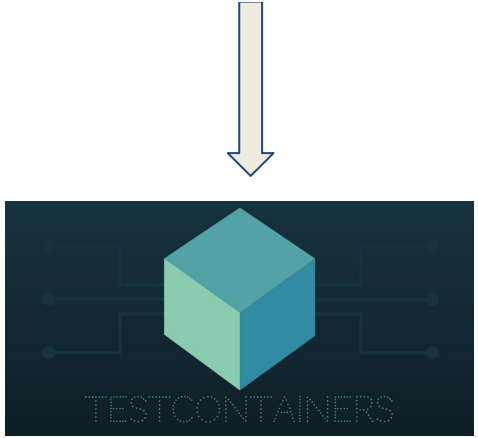
Spring Boot Test +
RestAssured + MockServer

```
@ContextConfiguration(classes = MockItTestConfiguration.class)
```

Before run tests

THE DEVELOPER'S CONFERENCE

```
@ContextConfiguration(classes = {DatabaseTestConfiguration.class, MockItTestConfiguration.class})
@SpringBootTest(webEnvironment = RANDOM_PORT)
@RunWith(SpringRunner.class)
class TranslatorIT {
```



MockServer

https://www.testcontainers.org/

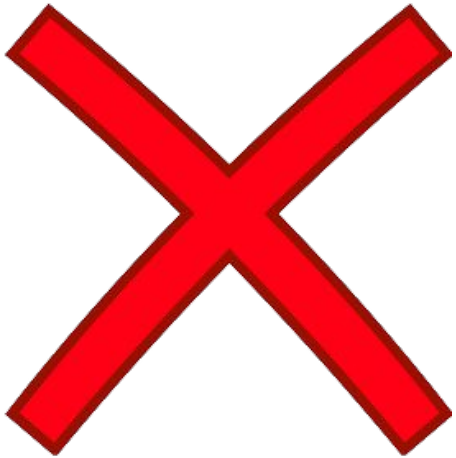
O que faz a biblioteca

THE DEVELOPER'S CONFERENCE

- Centraliza outras bibliotecas (starter)
- Faz a gestão das versões de terceiros
- Facilita o setup de um novo serviço
- Encapsula a infraestrutura

E?

Vamos checar?



O que a checagem resolve?

- **Problema:** É o que importa e pode estar escondido, separar da solução. Ex.: Vamos escrever testes poder fazer liberações em produção a qualquer momento. Problema real: **Não é possível liberar código em produção a qualquer momento.**

Antes de começar a checar

- **Comunicação:** Todos devem entender a importância do problema e a solução Ex.: dois desenvolvedores sabem que a maioria das liberações em produção afetam os clientes, mas o restante do time desconhece. **Comunicação + problema: alinhamento**

O que checar?

- **Meta/Medida:** Checagem deve ser o mais simples possível, Ex.: Testes escritos na semana, **valor + meta = resultado**.

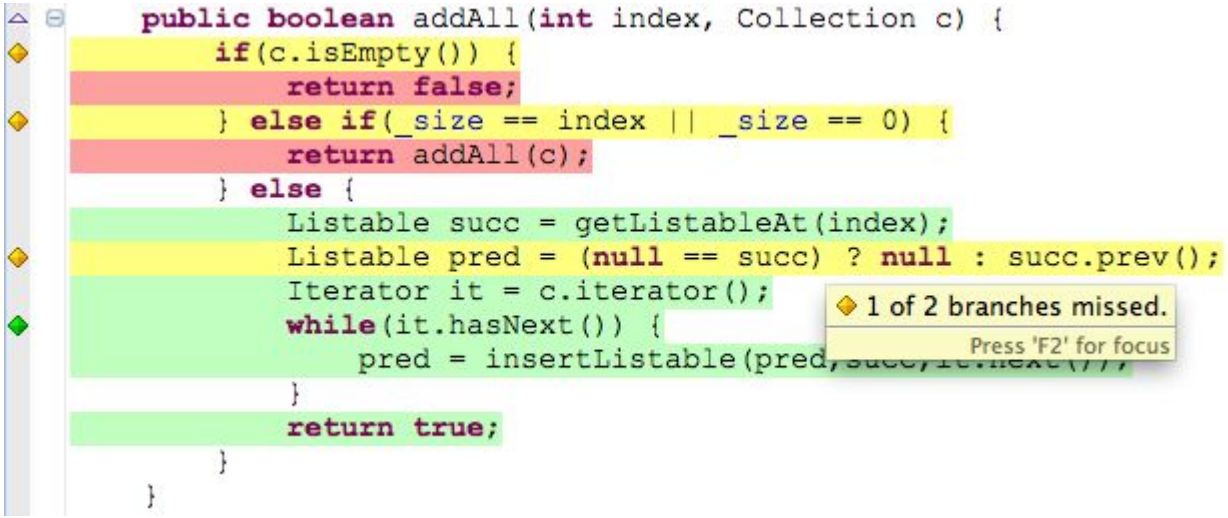
Vigilância

- **Monitoramento/Feedback:** Ex.: acordei em fazer testes para liberar código em produção livre de bugs, como saber se é estou sem checagem? **Posso liberar uma versão agora?**

Checagem quantitativa

THE DEVELOPER'S CONFERENCE

- **Cobertura de código:**



Como garantir minha evolução?

THE DEVELOPER'S CONFERENCE

- JaCoCo, baixou? O build quebra....

```
    <rule>
        <element>BUNDLE</element>
        <limits>
            <limit>
                <counter>LINE</counter>
                <value>COVEREDRATIO</value>
                <minimum>0.99</minimum>
            </limit>
            <limit>
                <counter>BRANCH</counter>
                <value>COVEREDRATIO</value>
                <minimum>0.99</minimum>
            </limit>
        </limits>
```

```
[INFO] --- jacoco-maven-plugin:0.8.3:check (check) @ jacoco-demo ---
[INFO] Loading execution data file /home/alisson/Code/jacoco-demo/targe
[INFO] Analyzed bundle 'jacoco-demo' with 1 classes
[WARNING] Rule violated for bundle jacoco-demo: lines covered ratio is 0.33, but expected minimum is 0.99
[INFO] ------------------------------------------------------------------------
[INFO] BUILD FAILURE
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 10.030 s
[INFO] Finished at: 2019-04-01T21:37:29-03:00
[INFO] ------------------------------------------------------------------------
[ERROR] Failed to execute goal org.jacoco:jacoco-maven-plugin:0.8.3:check (check) on project jacoco-demo: Cove
ee log for details. -> [Help 1]
```

Estou evoluindo, mas com bons testes?

THE DEVELOPER'S CONFERENCE

Qual código está melhor testado?

mutation

Element	Missed Instructions	Cov.	Missed Branches	Cov.
com.tdc.mutation.calculator.v1		100%		n/a
com.tdc.mutation.calculator.v2		90%		n/a
Total	3 of 62	95%	0 of 0	n/a

Cobertura pode ser uma cilada bino.

Mutação vs cobertura

THE DEVELOPER'S CONFERENCE

- Plugin de mutação, garantindo a qualidade!

Pit Test Coverage Report

Project Summary

Number of Classes	Line Coverage		Mutation Coverage	
2	90%	9/10	81%	13/16

Breakdown by Package

Name	Number of Classes	Line Coverage		Mutation Coverage	
com.tdc.mutation.calculator.v1	1	100%	5/5	63%	5/8
com.tdc.mutation.calculator.v2	1	80%	4/5	100%	8/8

Testes rasos = **mutação baixa**

THE DEVELOPER'S CONFERENCE

```java
@Test
public void constructor() { new CalculatorV1(); }

@Test
public void add() { Assertions.assertThat(CalculatorV1.add(1D, 2D)).isNotZero(); }

@Test
public void divide() { Assertions.assertThat(CalculatorV1.divide(1D, 2D)).isNotNull(); }
```

Mutations

```
1. Replaced double addition with subtraction → SURVIVED
2. mutated return of Object value for com/tdc/mutation/calcula
new RuntimeException ) → KILLED
1. Replaced double division with multiplication → SURVIVED
2. mutated return of Object value for com/tdc/mutation/calcula
new RuntimeException ) → KILLED
1. Replaced double subtraction with addition → SURVIVED
```

Testes com profundidade = **mutação alta**

THE DEVELOPER'S CONFERENCE

```java
@Test
public void subtract() {
    Assertions.assertThat(CalculatorV2.subtract(25D, 5D)) A
        .isNotNull() capture of ?
        .isPositive() capture of ?
        .isNotZero() capture of ?
        .isEqualTo(20D);
}

@Test
public void multiply() {
    Assertions.assertThat(CalculatorV2.m
        .isNotNull() capture of ?
        .isPositive() capture of ?
        .isNotZero() capture of ?
        .isEqualTo(127.5);
}
```

com.tdc.mutation.calculator.v2

Number of Classes	Line Coverage		Mutation Coverage	
1	80%	4/5	100%	8/8

Breakdown by Class

Name	Line Coverage	Mutation Coverage
CalculatorV2	80% 4/5	100% 8/8

```
[INFO] BUILD FAILURE
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  8.658 s
[INFO] Finished at: 2019-04-21T17:20:09-03:00
[INFO] ------------------------------------------------------------------------
[ERROR] Failed to execute goal org.pitest:pitest-maven:1.4.7:mutationCoverage
default) on project mutation: Mutation score of 81 is below threshold of 99
```

Como dar visibilidade e transparência?

THE DEVELOPER'S CONFERENCE

- Ferramentas como:
  - Code Climate
  - Sonar
  - Codacy

Que tal colocar uma dessas ferramentas nos seus Pull Request?

Show me the code

THE DEVELOPER'S CONFERENCE

- Sonar Cloud - Demo

THE DEVELOPER'S CONFERENCE

Obrigado!

THE DEVELOPER'S CONFERENCE

https://github.com/sandrogiacom/jtest-tools

https://github.com/sandrogiacom/testcontainers-demo

https://github.com/AlissonMedeiros/jacoco-demo

https://github.com/AlissonMedeiros/mutation-example