

Estratégias de testes em 10 passos, step by step!



THE
DEVELOPER'S
CONFERENCE 2019

Estratégias de testes na DevTest?

Trilha DevTest

Qualidade garantida ou seu bug de volta", você concorda? Se sim, essa trilha é para você, venha aprimorar seu mindset.

Qualidade de software é responsabilidade de todos, e para que isso ocorra com sucesso sua estratégia precisa ser definida conforme as necessidades de cada contexto.

Estratégias de testes na DevTest?

- Se a responsabilidade da qualidade é de todos, logo a ...
 - Estratégia
 - Conscientização de Impactos
 - Definição de Riscos
 - Cobertura

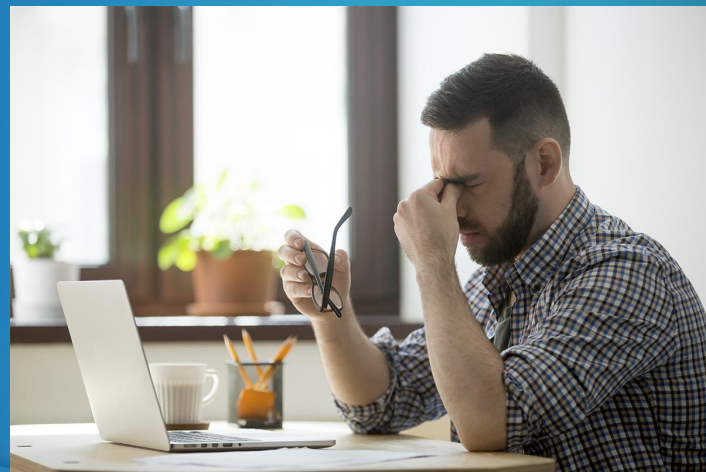
São responsabilidades de todos: Time Alinhado

Step by step...



1. Entendimento dos tipos de teste

- Conhecer os tipos de testes
- Aplicar de acordo com contexto
- Nível do Teste
- Ferramentas para cada tipo de teste



Analogia: Festa de Aniversário

Tipos
de
Teste



Teste Unitário

Tipos de Teste

- Valida um componente
- Menor unidade de código
- Independente



Tipos de Teste

Teste Integração

- Validar unidades trabalhando em conjunto
- Validar integração entre componentes



Tipos de Teste

Teste de Serviço

- Testes API
- Contrato
- Requisição
- Resposta
- Persistência



Tipos de Teste

Teste UI

- Validar o front-end



Tipos de Teste

Teste E2E

- Cenários mais próximos dos reais (PROD)
- Ponta a ponta
- Jornada do Usuário
- Contempla todas integrações



Tipos de Teste

Teste Exploratório

- Exercitar cenários\fluxos diferentes no meio do caminho feliz
- Pensar em fluxos alternativos, outras possibilidades



Tipos de Teste

Teste Manual

- Mais conhecidos por nós
- Fluxos são validados manualmente



Tipos de Teste

Teste Não Funcional

- **Performance** - Medir a capacidade e o desempenho do processamento do software diante de alto volume de transações\usuários trabalhando simultaneamente (**Aplicação Escalável**)

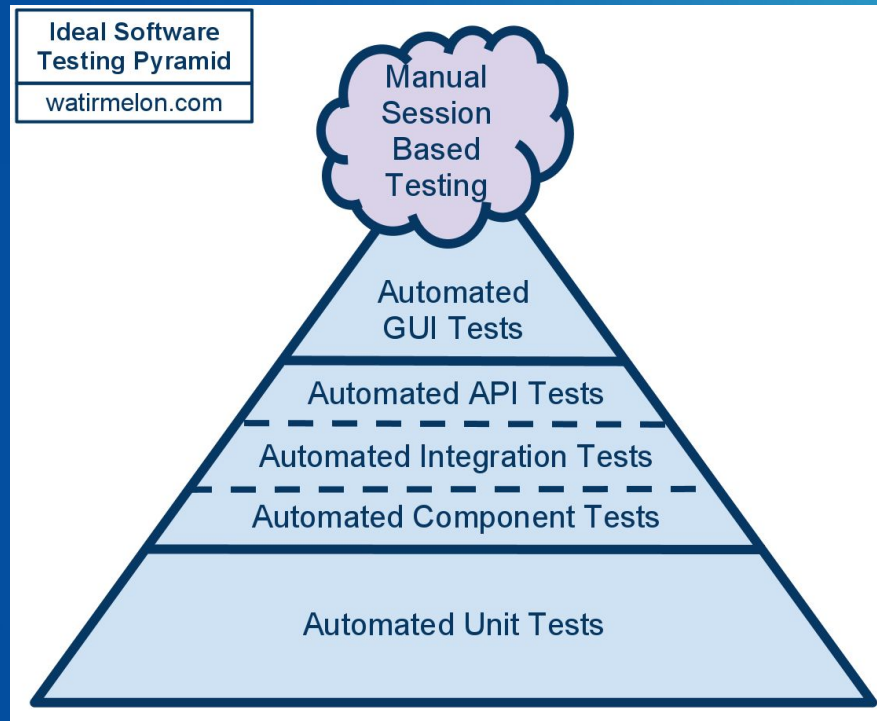


Teste Não Funcional

- **Segurança**
 - Verificar as vulnerabilidades frente a diferentes ataques nas aplicações\serviços
- **Usabilidade**
 - Validar sobre perspectiva do usuário
 - Facilidade de manuseio\operação\feedback para o usuário
- **Acessibilidade**
 - Validar normas básicas de acessibilidade para da aplicação por usuários com deficiência física



2. Aplicar a Pirâmide





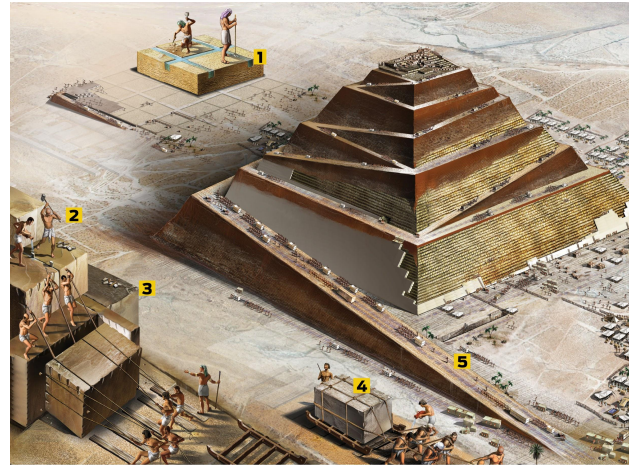
INDISCHE S.
MADAGASCAR
POLAR-KREIS



POLAR-KREIS
d Pol
obischer

Estratégias de Teste A pirâmide

- Qual camada faz mais sentido o teste ser executado?
- Qual tipo de teste aplicar?
- Considerar complexidade\relevância



3. Identificar Vantagens

- Minimizar retrabalho
- Feedbacks mais rápidos
- Estreita a relação\ aumenta o nível de confiança do time

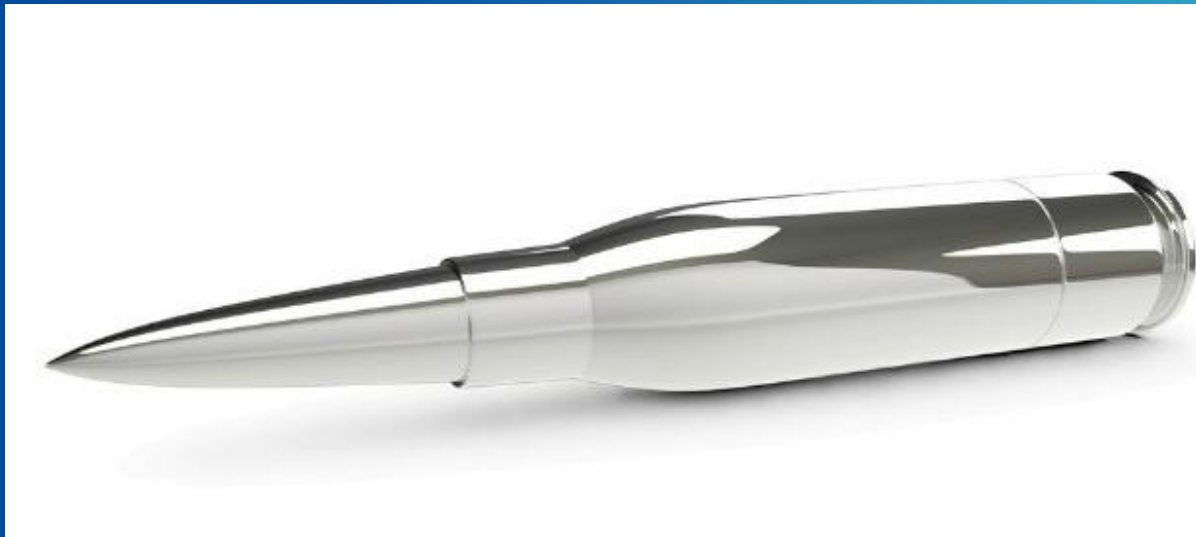


4. Identificar Desvantagens

- Como seremos impactados com essa abordagem?



5. Não é bala de prata



6. Distribuir os Testes

Camadas

- Ganhar escala
- Ponderar **Cobertura** x Valor para Negócio



7. Integrar - Não Replicar

- Não replicar o mesmo teste em camadas diferentes



8. Alinhar entendimento com time

- Em qual camada faz sentido o teste ser executado?
- Qual tipo de teste aplicar?
- Considerar
 - **Complexidade**
 - **Relevância**



9. Arquitetura\Boas Práticas

- Dica mais importante

➔ **Base da sua arquitetura olhando sempre manutenção**

- Convenção de Nomes
- **Reuso:** Encapsulamento
- DSL's
- Legibilidade - **Clean Code**

```
11 @Test
12 public void validaTaxaParaContaPessoaFisicaTest() {
13     ContaCorrente contaTeste = new ContaCorrente("Fulano",
14                                                     1234,
15                                                     123456789,
16                                                     1);
17
18     assertEquals(contaTeste.getTaxa(), 20.00);
19 }
20
21
22 @Test
23 public void validaTaxaParaContaPessoaJuridicaTest() {
24     ContaCorrente contaTeste = new ContaCorrente("Beltrano",
25                                                     1234,
26                                                     123456789,
27                                                     2);
28
29     assertEquals(contaTeste.getTaxa(), 15.00);
30 }
31
```

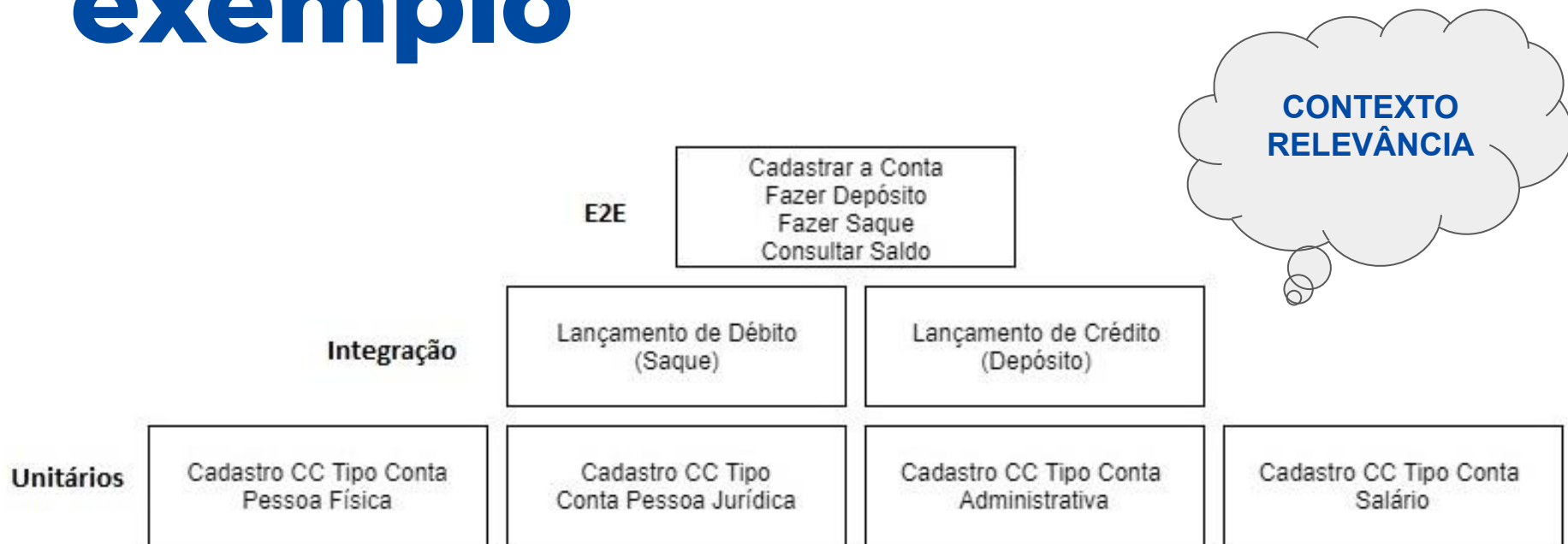
```
33 @Test
34 public void validaTaxaParaContaAdministrativaTest() {
35     ContaCorrente contaTeste = new ContaCorrente("Ciclano",
36                                                     1234,
37                                                     123456789,
38                                                     3);
39
40     assertEquals(contaTeste.getTaxa(), 100.00);
41 }
42
43
44 @Test
45 public void validaTaxaParaContaSalarioTest() {
46     ContaCorrente contaTeste = new ContaCorrente("Alguém",
47                                                     1234,
48                                                     123456789,
49                                                     4);
50
51     assertEquals(contaTeste.getTaxa(), 5.00);
52 }
53
```

```
11
12 public class ValidaTaxaParaContaTest {
13
14     ContaCorrenteDsl dsl = ContaCorrenteDsl.instanciaDsl();
15
16     @DataProvider(name = "massaDeDados")
17     public Object[][] criacaoMassaDados() {
18         return new Object[][] {
19             // Tipo Conta           Taxa           Cenario
20             { TipoConta.PESSOA_FISICA , 20.00 , "Valida taxa para conta corrente pessoa física" },
21             { TipoConta.PESSOA_JURIDICA , 15.00 , "Valida taxa para conta corrente pessoa jurídica"},
22             { TipoConta.ADMINISTRATIVA , 100.00 , "Valida taxa para conta corrente administrativa" },
23             { TipoConta.SALARIO , 5.00 , "Valida taxa para conta corrente salario" }
24         };
25     }
26
27     @Test (dataProvider = "massaDeDados")
28     public void validaTaxaParaContaTest( TipoConta tipoConta,
29                                         double taxa,
30                                         String cenario) {
31
32
33         ContaCorrente contaTeste = dsl.criaConta(tipoConta.getCodigo());
34
35         assertEquals(contaTeste.getTaxa(), taxa);
36     }
37 }
```

Estrutura DSL

```
5 public class ContaCorrenteDsl {
6
7     private ContaCorrenteDsl() {
8
9     }
10
11     public static ContaCorrenteDsl instanciaDsl() {
12         return new ContaCorrenteDsl();
13     }
14
15     public ContaCorrente criaConta(int tipo) {
16         return new ContaCorrente("Teste Conta Corrente",
17                                 1234,
18                                 123456789,
19                                 tipo);
20     }
21
22     public ContaCorrente criaContaComSaldo(double valorSaldo, int tipo) {
23         ContaCorrente conta = new ContaCorrente("Teste Conta Corrente",
24                                                 1234,
25                                                 123456789,
26                                                 tipo);
27
28         conta.fazDeposito(valorSaldo);
29
30         return conta;
31     }
32
33 }
```


Pirâmide de Testes do exemplo



Estratégias de Teste

10. Go

- É preciso que se torne dentro do time uma ...



Cultura x Enxergar valor



FOTORESEARCH®



Ariane Izac



Analista de Testes
Há 12 anos



Matera Systems
Há 7 anos
Blogueira



Grupo no LinkedIn
Diário de uma Paixão:
Teste de Software



CONTATOS

Linkedin: **Ariane Izac** Email: **afizac@gmail.com** Twitter: **@arianizac**



LET'S **CREATE THE FUTURE**