

# Teste de Software



Prof. Marcos Rodrigo momo, M.Sc.  
[marcos.momo@ifsc.edu.br](mailto:marcos.momo@ifsc.edu.br)

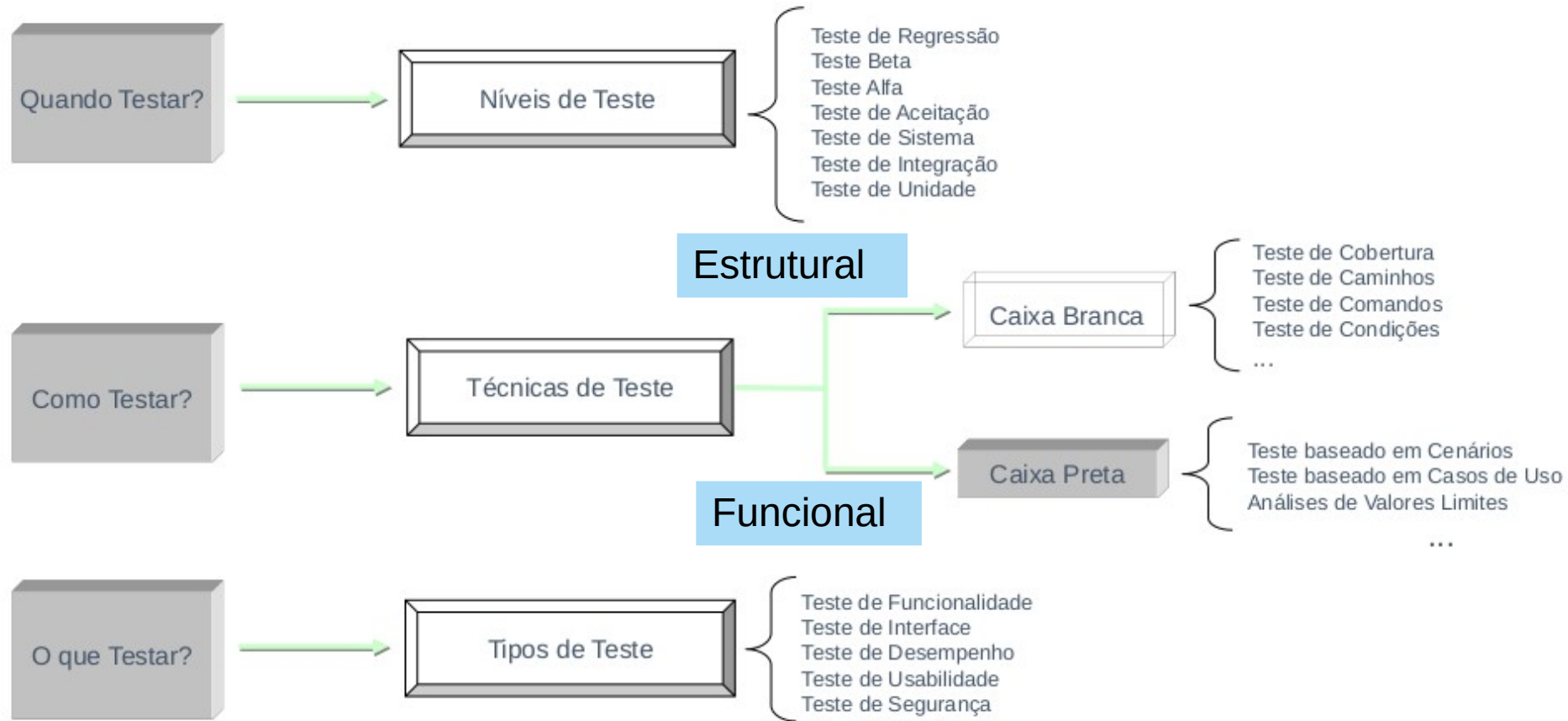
Gaspar, maio 2021.

# Técnicas Estáticas de Teste

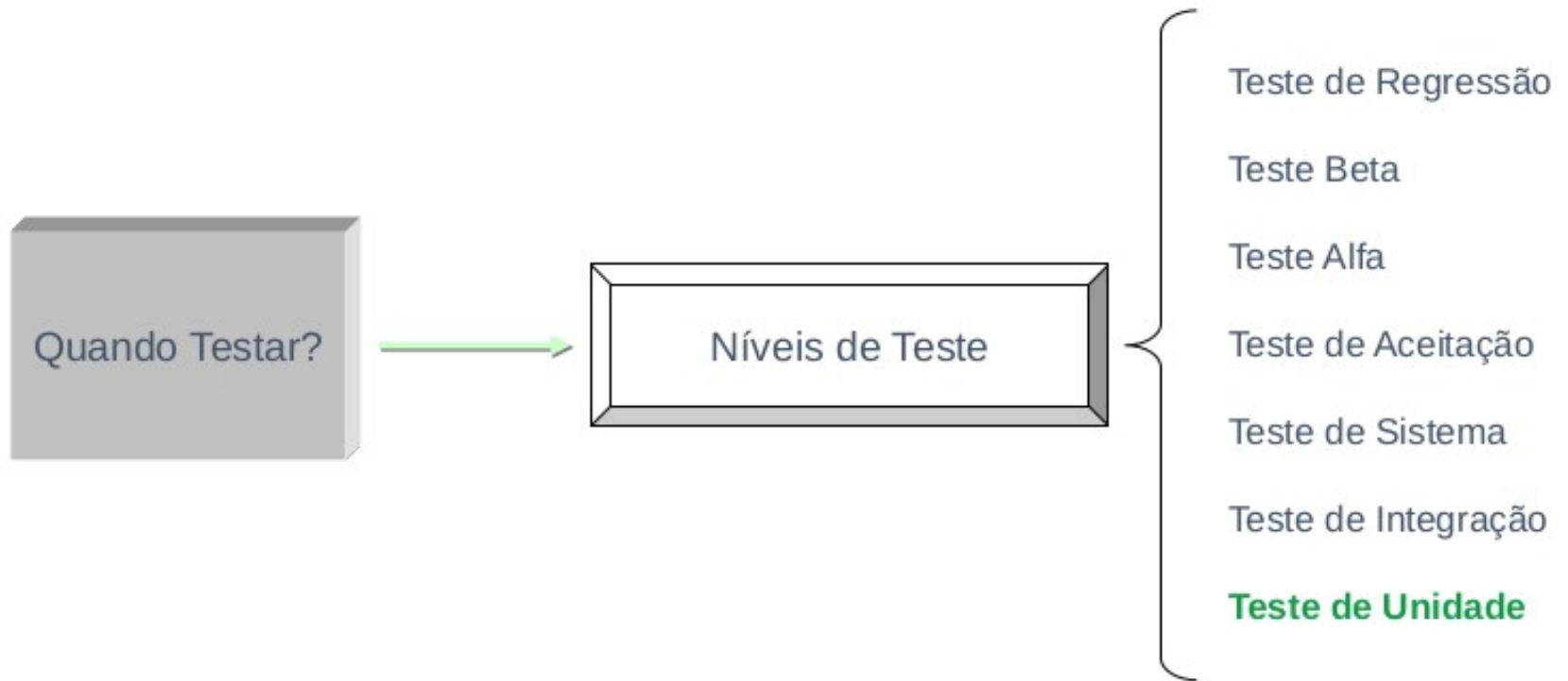


- Revisão e o Processo de Teste
- Processo de Revisão
  - Fases de uma Revisão Formal
  - Papéis e Responsabilidades em Revisão
  - Tipos de Revisão
  - Fatores de sucesso para Revisão
- Ferramentas de Análise Estática
- Revisão teórica
- Trabalho final

# Fundamentos do teste de software



# Níveis de teste

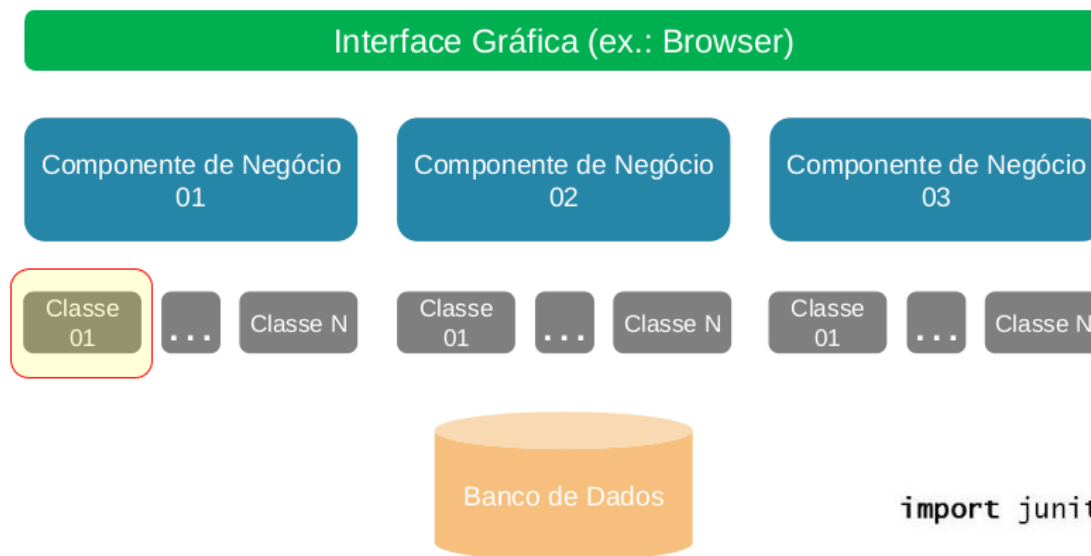


# Teste de unidade



- **Objetivo:**
  - encontrar falhas de funcionamento dentro de uma pequena parte do sistema funcionando independentemente do todo
- Feitos pelo programador
- O alvo são subrotinas, métodos, classes
  - Exemplo, as menores unidades do sistema
- Geralmente é automatizado, através de ferramentas como **Junit**, PHPUnit, XXXUnit e outras
- Precisa estar sempre atualizado, coerente com as regras de negócio atuais do sistema

# Teste de unidade



```
import junit.framework.TestCase;
```

```
/**
```

```
 * Testes de unidade para a classe {@link Pessoa}
```

```
 */
```

```
public class TestPessoa extends TestCase {
```

```
/**
```

```
 * Um teste de unidade para verificar se o nome está
```

```
 * formatado corretamente
```

```
 */
```

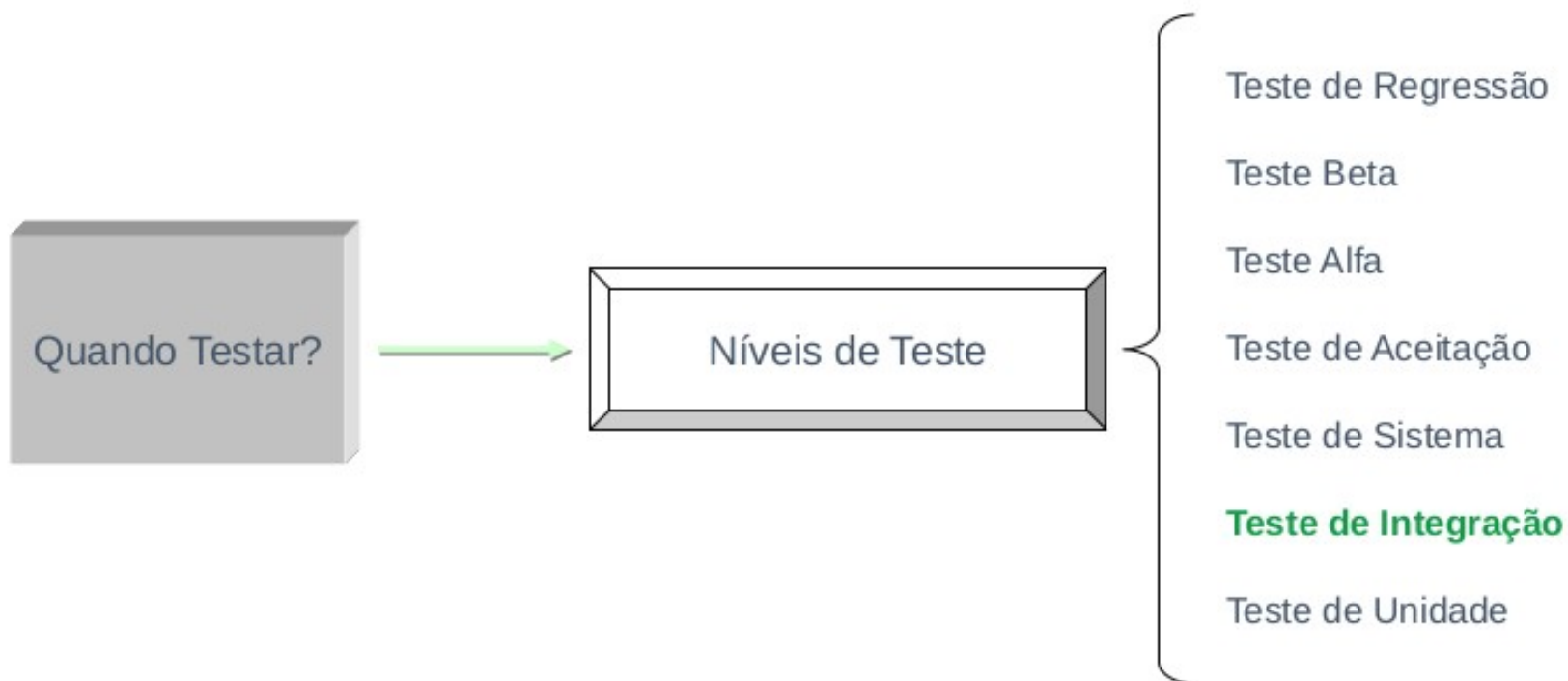
```
public void testObterNomeCompleto() {
```

```
    Pessoa p = new Pessoa("Fulano", "Tal");
```

```
    assertEquals("Fulano Tal", p.getNomeCompleto());
```

```
}
```

# Níveis de teste



# Teste de integração



- **Objetivo:**
  - validar a comunicação entre os componentes de um sistema
- Feitos pelo programador
- O alvo são funcionalidades que envolvem a integração de componentes
- Geralmente os tipos de falhas encontradas são de transmissão de dados
  - Exemplo, um componente A invoca um método de um componente B espera um valor inteiro, porém, vem um valor decimal, causando uma falha no componente A
- Geralmente é automatizado, através de ferramentas como **Junit**, PHPUnit, XXXUnit e outras
- Podem ser feitos antes de o sistema estar concluído, à medida em que os componentes vão ficando prontos





# Teste de integração



Interface Gráfica (ex.: Browser)

Componente de Negócio  
01

Componente de Negócio  
02

Componente de Negócio  
03

Classe  
01

...

Classe N

Classe  
01

...

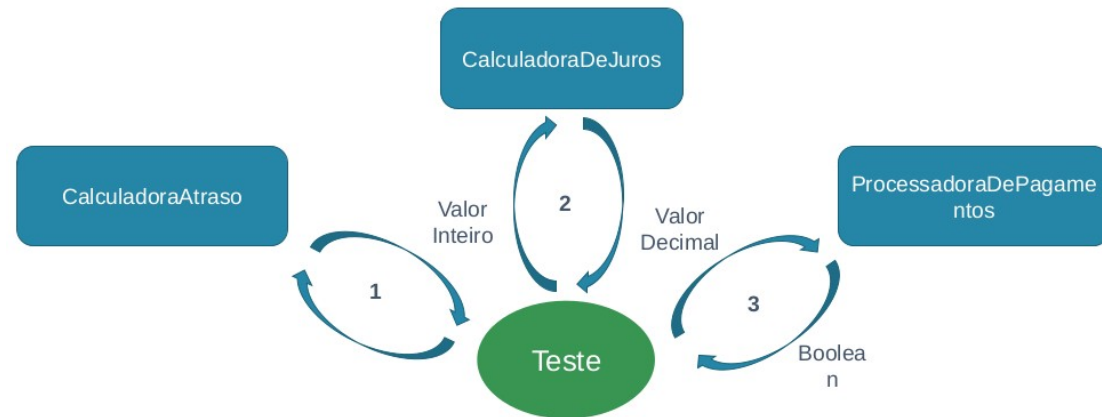
Classe N

Classe  
01

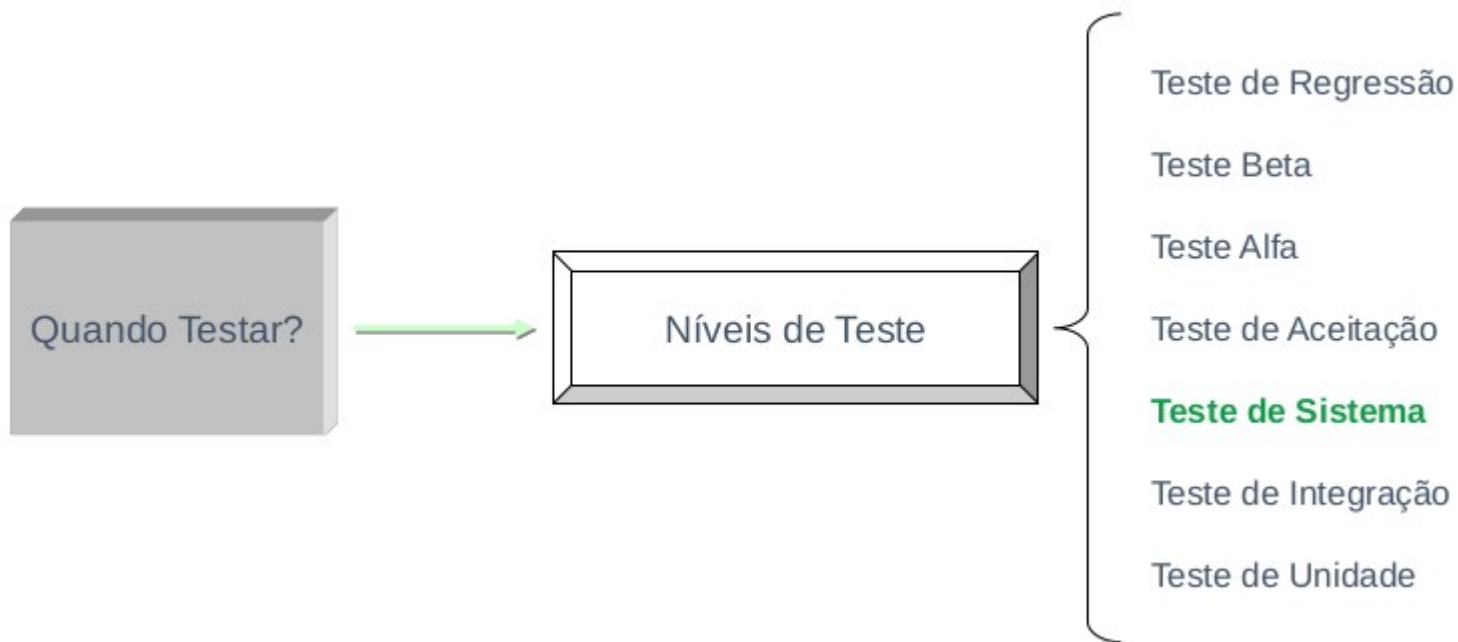
...

Classe N

Banco de Dados



# Níveis de teste





- **Objetivo:**
  - Executar o sistema sob ponto de vista de seu usuário final, varrendo as funcionalidades em busca de falhas em relação aos objetivos originais
- Planejados e executados pela equipe de teste
- Cenários de teste coerentes com os requisitos especificados para o sistema
- São realizados após a codificação do sistema estar concluída

# Teste de sistema



Interface Gráfica (ex.: Browser)

Componente de Negócio  
01

Componente de Negócio  
02

Componente de Negócio  
03

Classe  
01

...

Classe N

Classe  
01

...

Classe N

Classe  
01

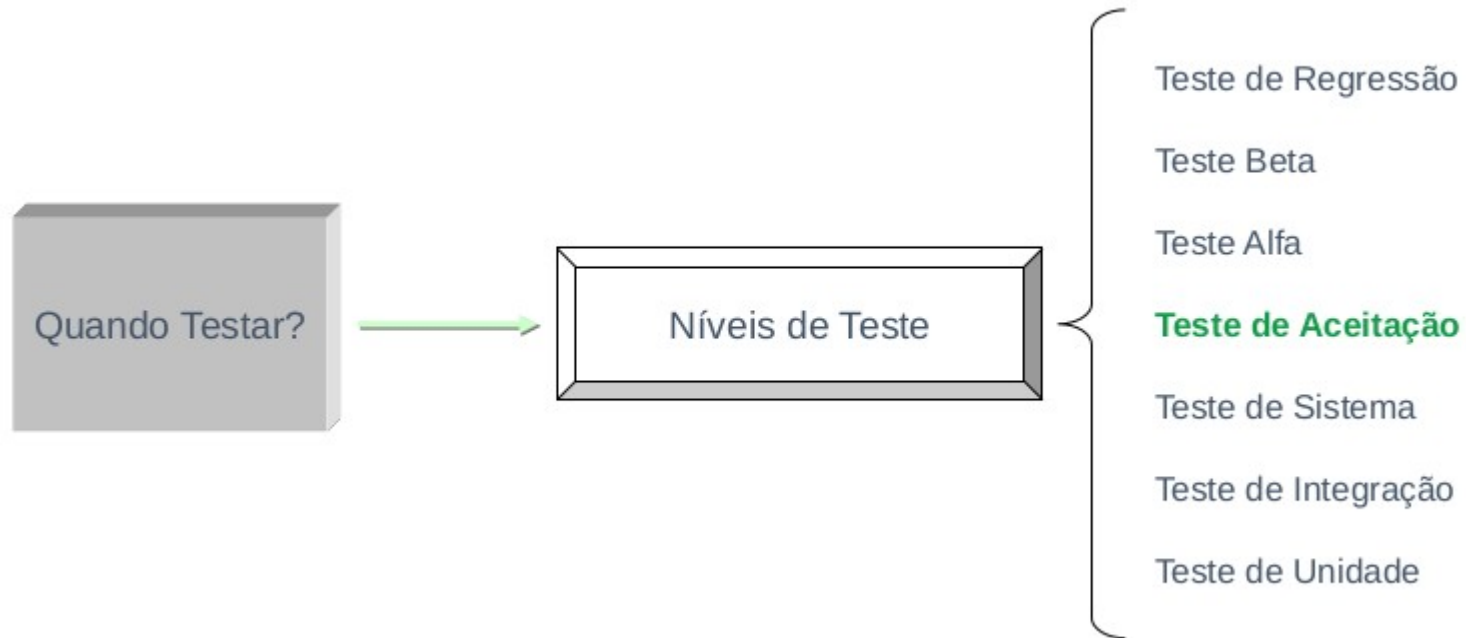
...

Classe N

Banco de Dados



# Níveis de teste





# Teste de aceitação



- **Objetivo:**
  - Executar o sistema sob ponto de vista de seu usuário final, varrendo as funcionalidades em busca de falhas em relação aos objetivos originais
- Planejados e executados por um grupo restrito de usuários finais do sistema, que simulam operações de rotina do sistema de modo a verificar se seu comportamento está de acordo com o solicitado
- Visa permitir ao cliente determinar se aceita ou não o sistema
- Pode incluir testes funcionais, de recuperação de falhas, de segurança e de desempenho

# Teste de aceitação



Interface Gráfica (ex.: Browser)

Componente de Negócio  
01

Componente de Negócio  
02

Componente de Negócio  
03

Classe  
01

...

Classe N

Classe  
01

...

Classe N

Classe  
01

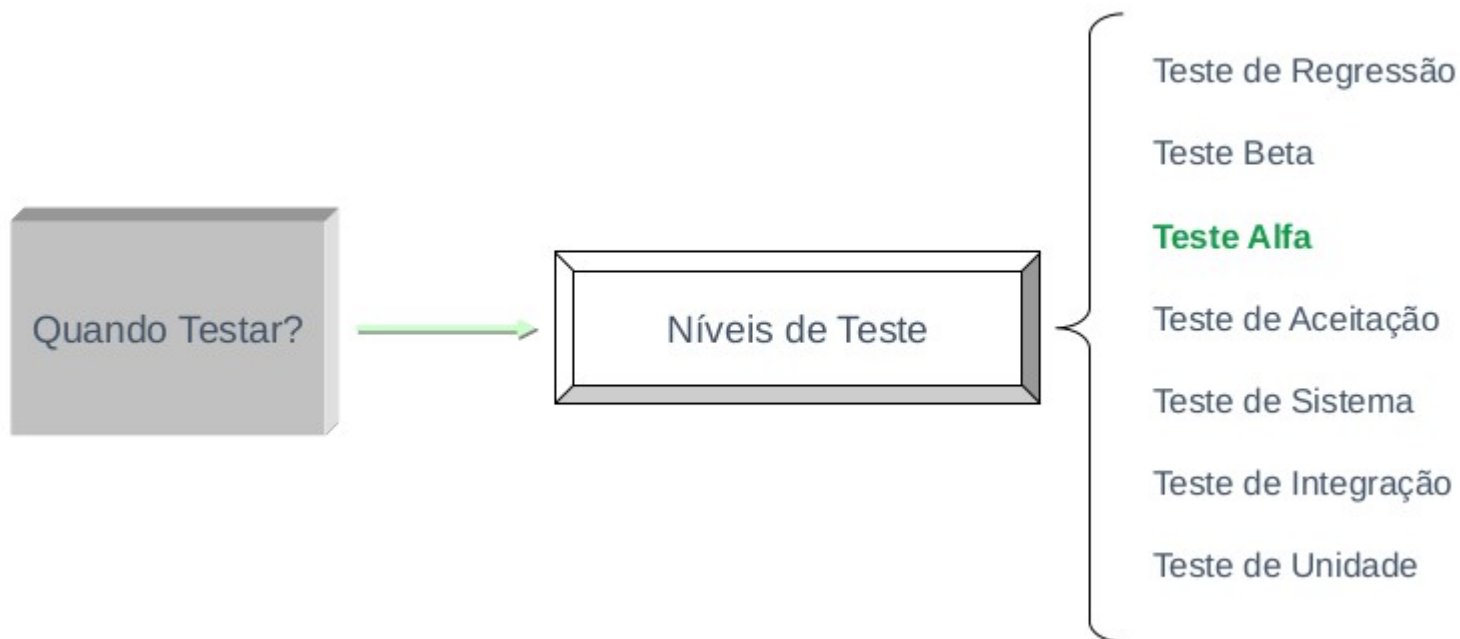
...

Classe N

Banco de Dados



# Níveis de teste







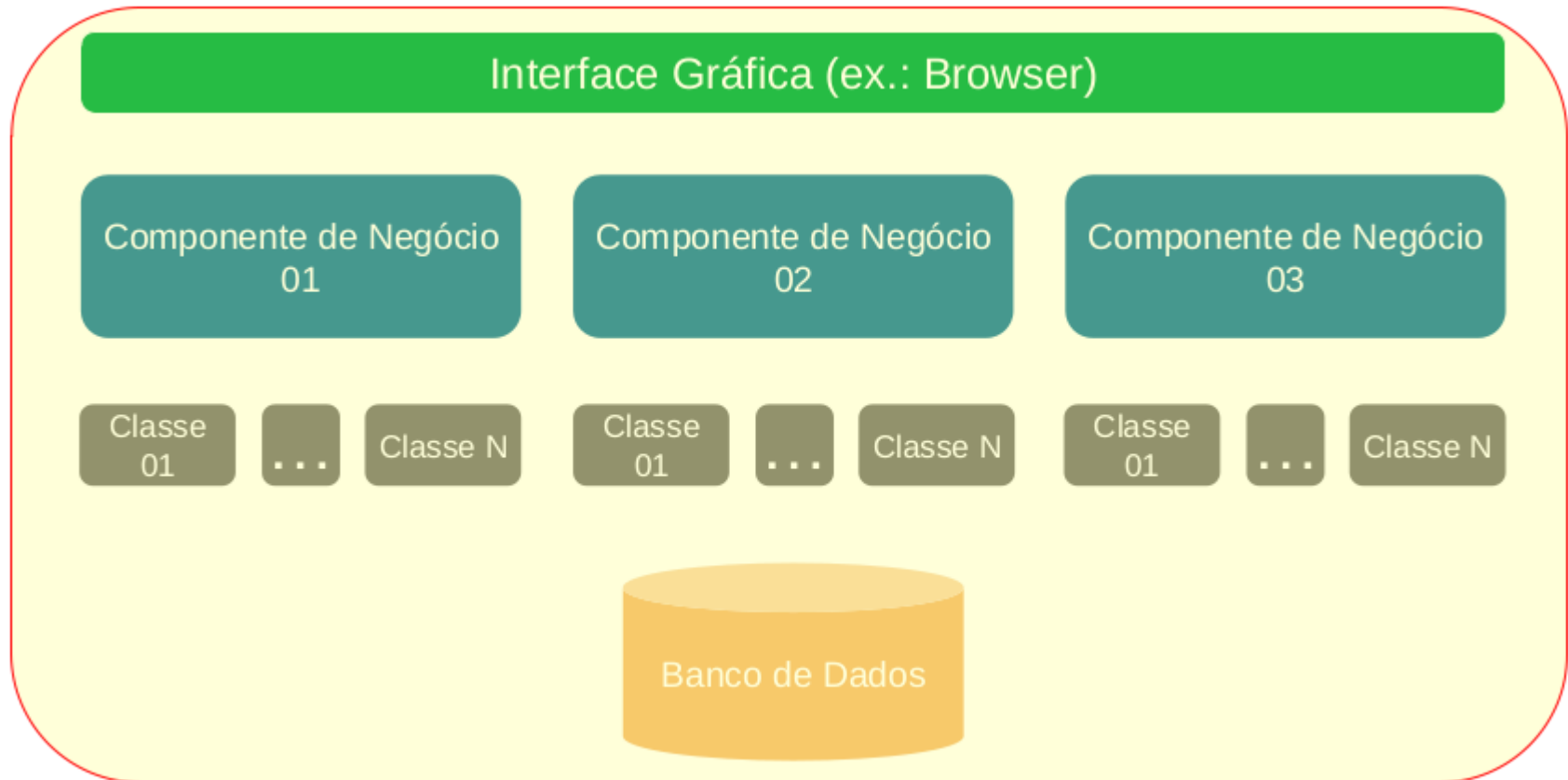
# Teste alfa



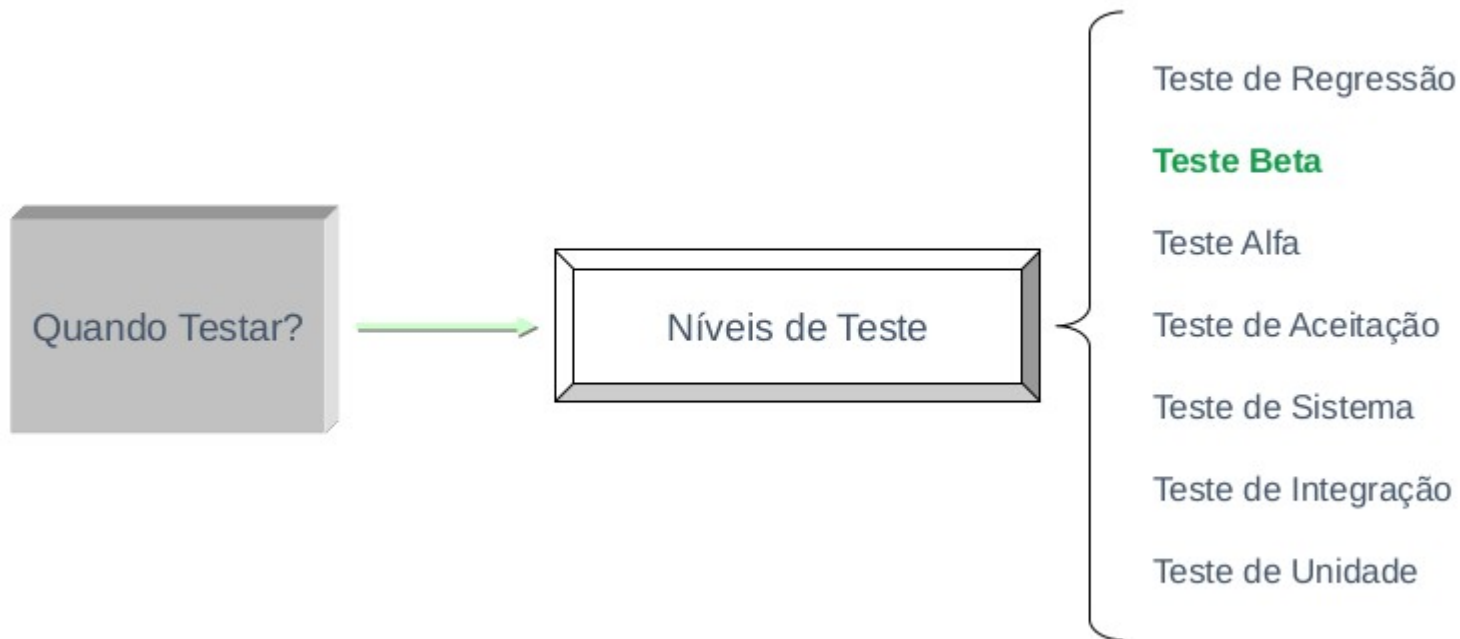
- **Objetivo:**

- Executar o sistema de forma não planejada, sob ponto de vista de seu usuário final, porém, apenas por um grupo pequeno de pessoas
- O grupo restrito de usuários que testarão geralmente é composto por membros da própria organização e também do cliente
- Visa a identificação de possíveis erros não detectados até o momento, encontrados enquanto mais usuários finais utilizam o sistema de forma natural, não planejada
- Representantes do time de programadores irão acompanhar de perto estes testes para coletar possíveis falhas a serem corrigidas e melhorias a serem implementadas

# Teste alfa



# Níveis de teste





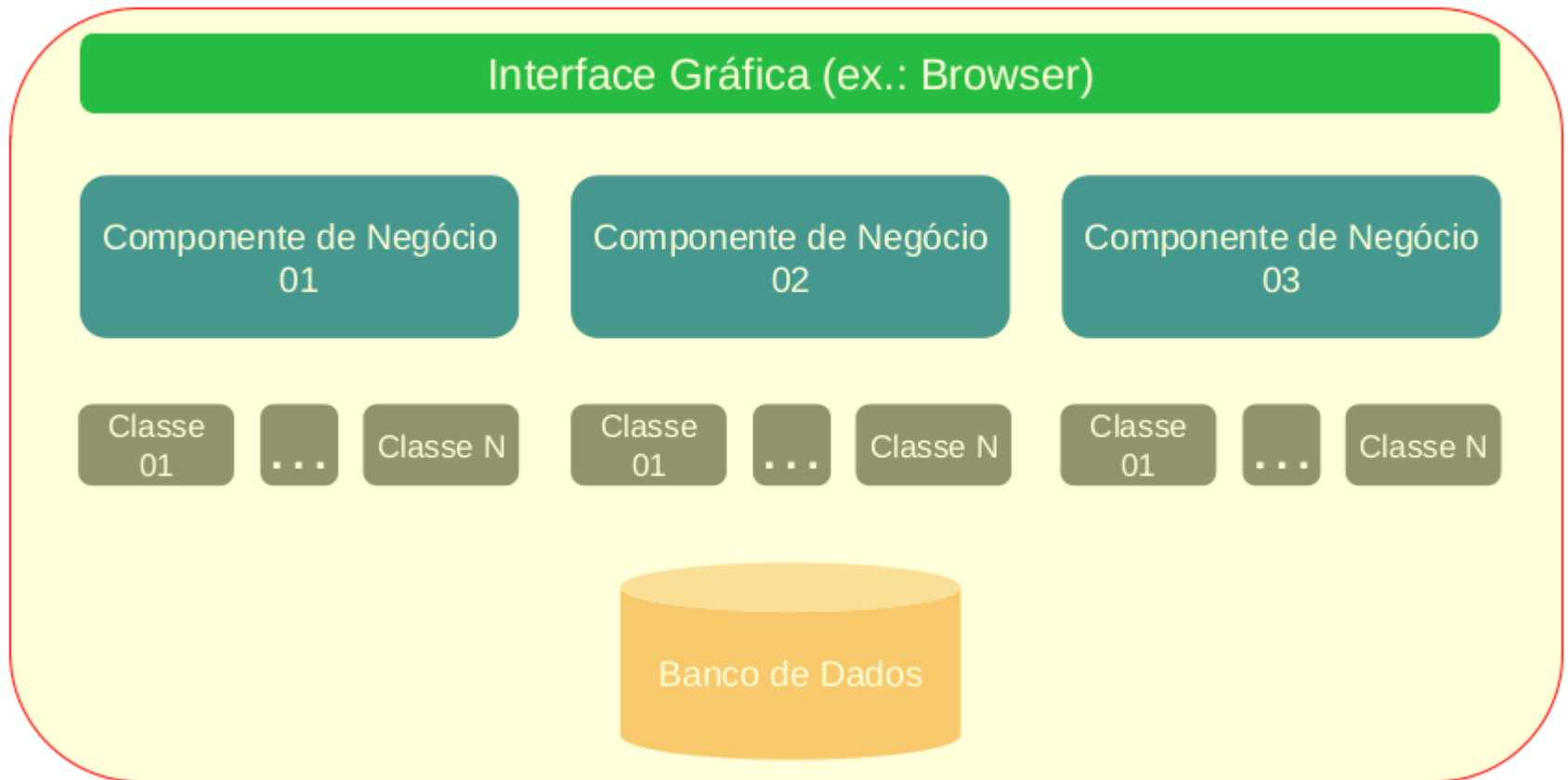
# Teste beta



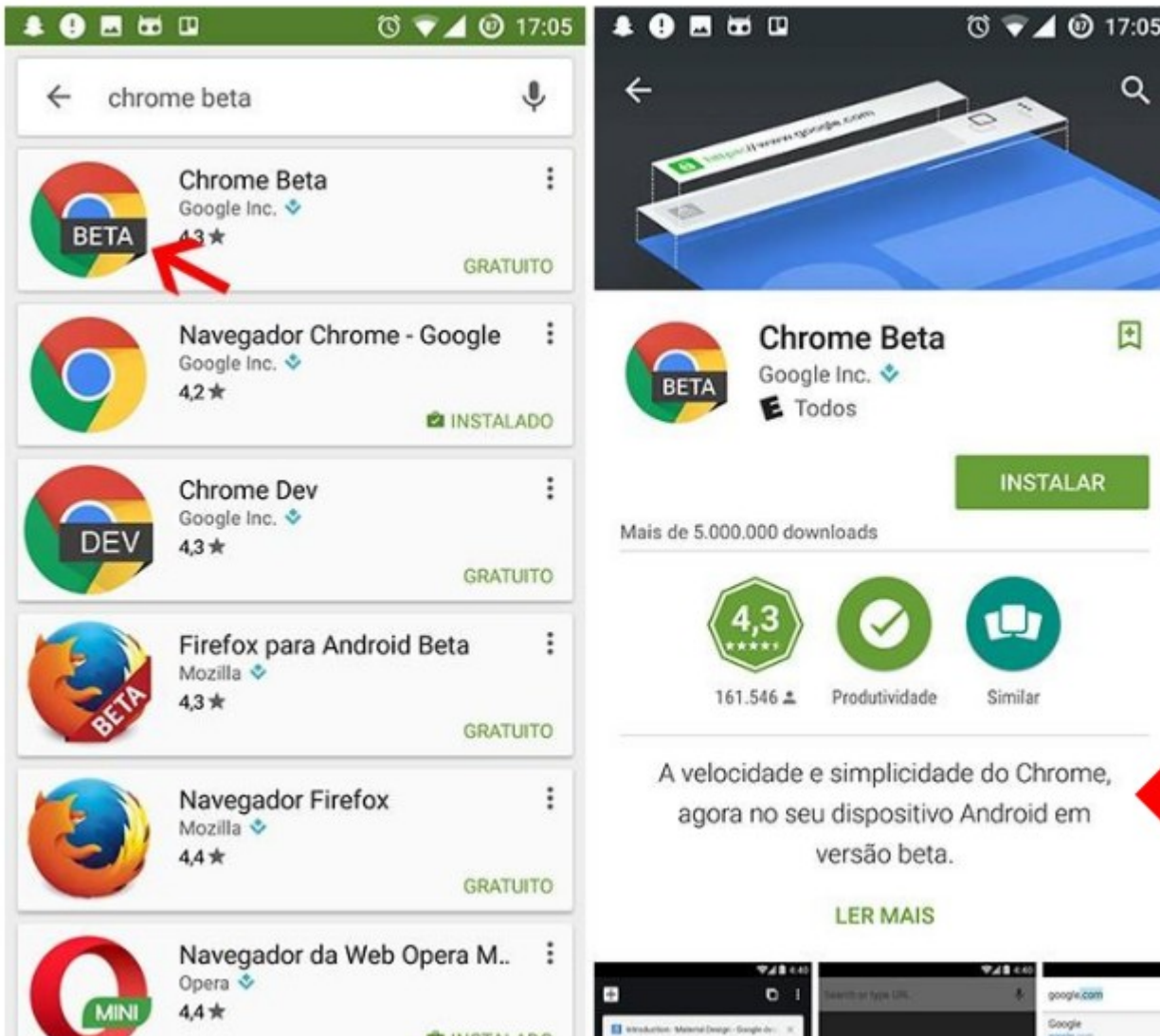
- **Objetivo:**

- Executar o sistema de forma não planejada, sob ponto de vista de seu usuário final, porém, por um grupo grande de pessoas
- O grupo de usuários que testarão geralmente são usuários reais e desconhecidos, sendo de uma determinada localidade, idioma ou que satisfizeram determinados critérios definidos pelo fornecedor do sistema
  - Exemplo, versão Beta do sistema ser lançada apenas em países cujo idioma é inglês
- Visa a identificação de possíveis erros não detectados até o momento, encontrados enquanto ainda mais usuários finais utilizam o sistema de forma natural, não planejada
- Representantes do time de programadores não irão fazer acompanhamento e coleta de erros. No teste Beta, os usuários é que reportam os erros encontrados

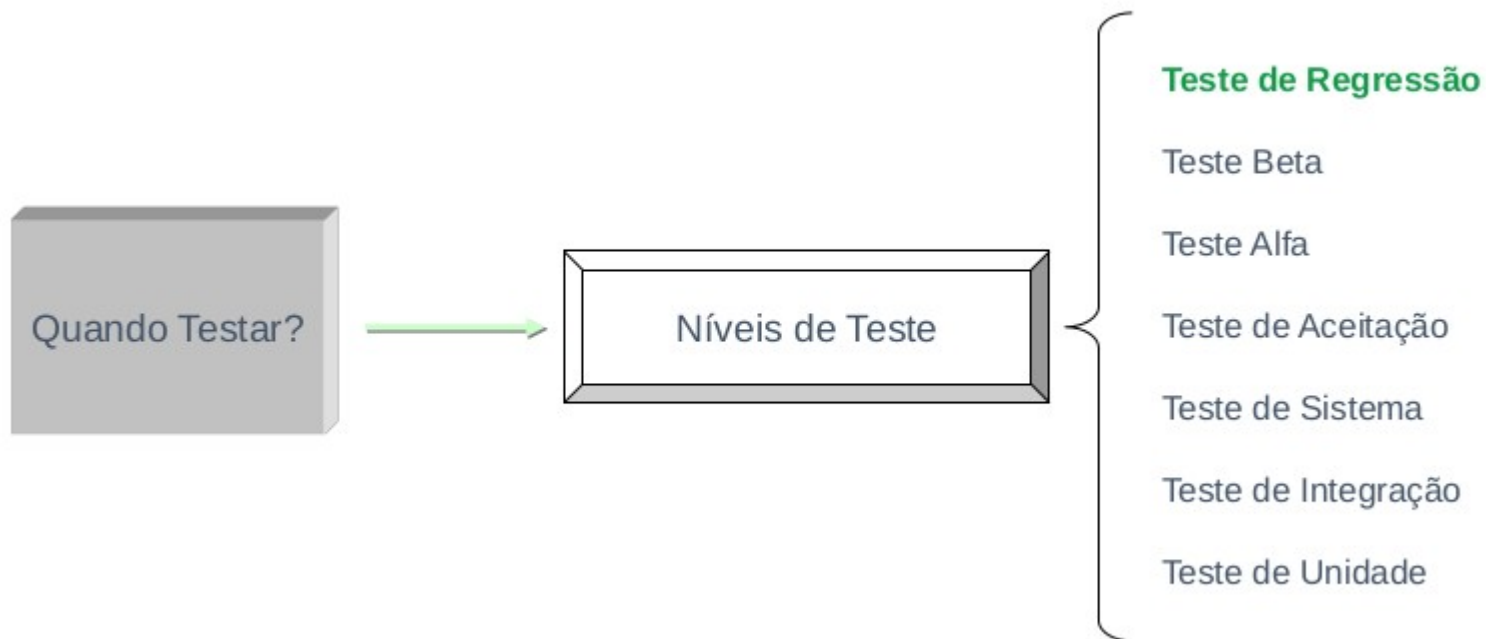
# Teste beta



# Teste beta



# Níveis de teste





# Teste de regressão



- **Objetivo:**

- Reexecutar testes após alterações serem realizadas no sistema, para conferir se tudo continua funcionando corretamente (detectar efeitos colaterais):
- Após mudanças de regras durante o desenvolvimento
  - Exemplo, facebook agora permite a postagem de vídeos 3D na timeline
- Após a correção de uma falha encontrada
- Após a implementação de melhorias para lançamento de novas versões
- Consiste em se aplicar, a cada nova versão do software ou a cada ciclo, todos os testes que já foram aplicados nas versões ou ciclos de teste anteriores do sistema
- É nessa hora que os testes automáticos fazem a maior diferença



# Teste de regressão



Interface Gráfica (ex.: Browser)

Componente de Negócio  
01

Componente de Negócio  
02

Componente de Negócio  
03

Classe  
01

...

Classe N

Classe  
01

...

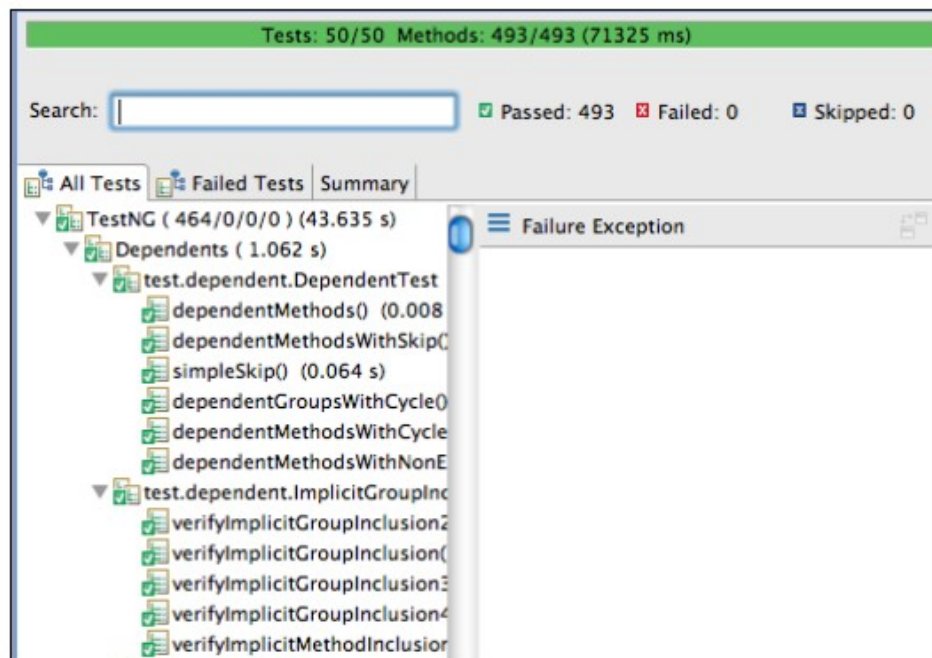
Classe N

Classe  
01

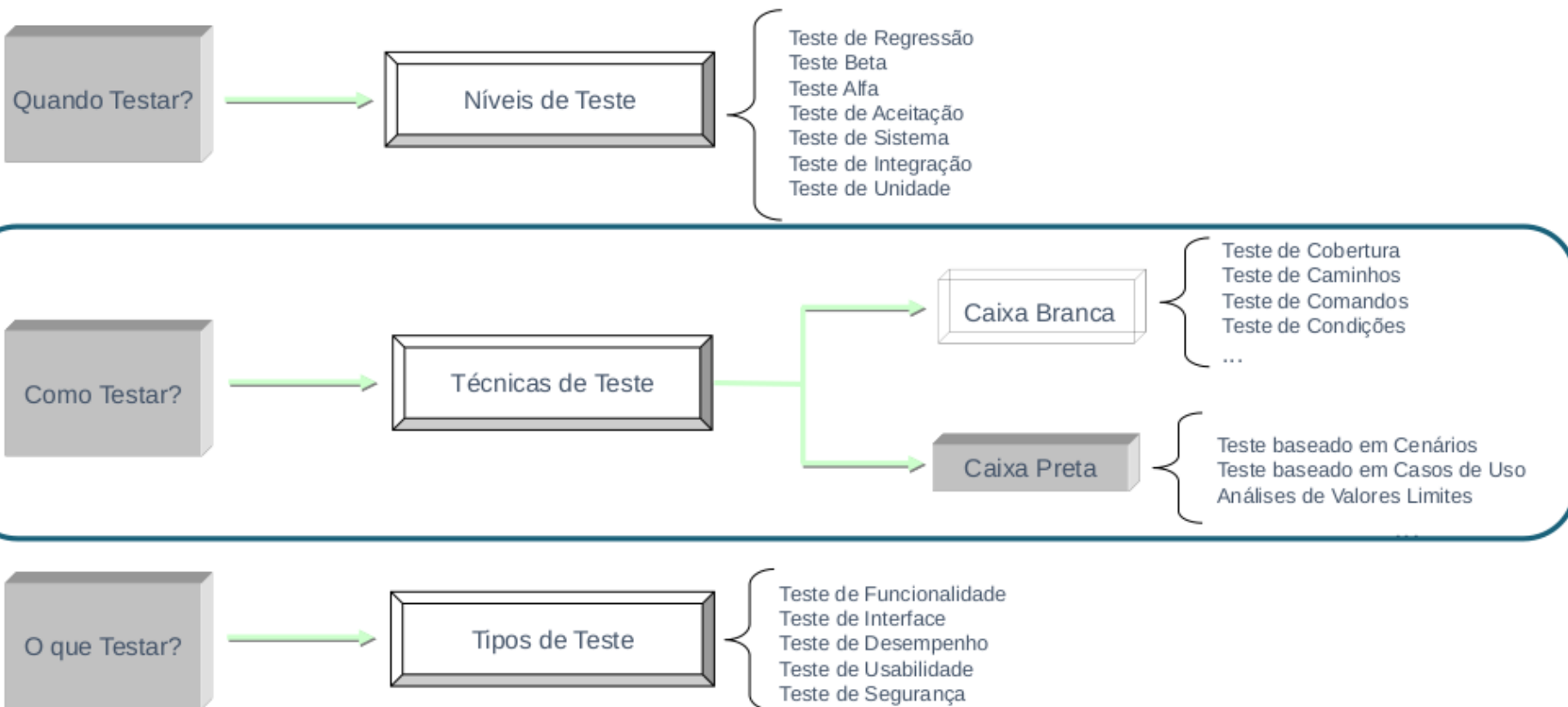
...

Classe N

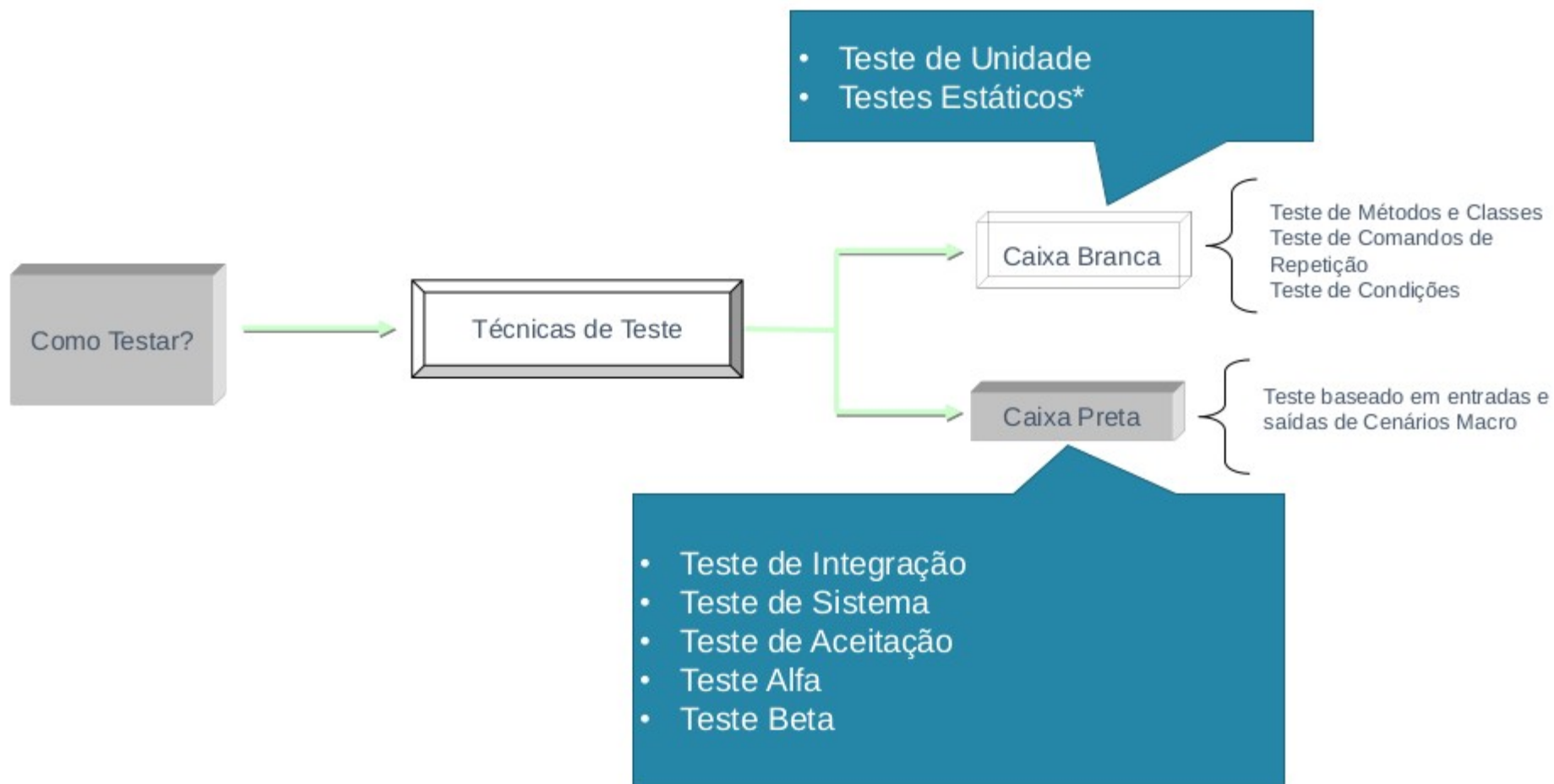
Banco de Dados



# Técnicas de teste



# Técnicas de teste



# Teste estático



- Esse teste consiste em apenas analisar o código e encontrar erros visível aos olhos
- Não executa o sistema para realizar o teste
- Principais erros encontrados:
  - Boas práticas de programação
  - Documentação de código
  - Tratamento de erros

# Teste Caixa Branca - Estrutural



```
1 package login;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.ResultSet;
6 import java.sql.Statement;
7
8 public class User {
9     public Connection conectarBD(){
10         Connection conn = null;
11         try{
12             Class.forName("com.mysql.Driver.Manager").newInstance();
13             String url = "jdbc:mysql://127.0.0.1/test?user=lopes&password=123";
14             conn = DriverManager.getConnection(url);
15         } catch (Exception e) { }
16         return conn;
17     }
18     public String nome="";
19     public boolean result = false;
20     public boolean verificarUsuario(String login, String senha){
21         String sql = "";
22         Connection conn = conectarBD();
23         //INSTRUÇÃO SQL
24         sql += "select nome from usuarios ";
25         sql += "where login = " + "'" + login + "'";
26         sql += " and senha = " + "'" + senha + "'";
27         try{
28             Statement st = conn.createStatement();
29             ResultSet rs = st.executeQuery(sql);
30             if(rs.next()){
31                 result = true;
32                 nome = rs.getString("nome");
33             } catch (Exception e) { }
34             return result; }
35     } //fim da classe
```

## Análise Estática:

- Não documentado
- Sem tratamento de exceções
- Não fecha conexão após utilizá-la

# Teste Caixa Preta - Funcional



proj4 me

Gustavo Farias

Minha Organização | Contatos | Projetos | Relatórios | Avaliação de Desempenho

Pesquisar

Dados Cadastrais

Config. Organização
















**Colaboradores**

Colaboradores

Nome:

Pesquisar

+ Cadastrar Novo Colaborador

Colaboradores Cadastrados		
Nome	Cargo	
Gustavo Farias	Project Manager	  
Jason Reid	Chief of Operation Office	  
Ana Sarmento	General Manager - City	  
Walter Cruz	Chief Executive Officer	  
Marta Martins	Customer Service Supervisor	  

<< < 1 2 > >> 5 (1 of 2)



- Farias, Gustavo. Automação de teste de software. Udemy. 2020.
- BSTQB. Base de Conhecimento para Certificação em Teste – Syllabus
- D. Grahan; V. Veenendaal; I. Evans; R. Black. Foundations of Software Testing: ISTQB Certification. Cengage Learning Business Press.
- Bastos, A.; Rios, E.; Cristalli, R. & Moreira, T. Base de conhecimento em teste de software.
- Justo, Daniela Sbizera Prof<sup>a</sup> Dra. Introdução a teste de software. IFSC. Gaspar.
- Anne Caroline O. Rocha – Tester Certified – BSTQB – NTI|UFPB ISTQB. Certified Tester Foundation Level Syllabus: Capítulo 3. Ago. 2011. Acesso em: 12/09/2018. Disponível em: [http://bstqb.org.br/uploads/syllabus/syllabus\\_ctfl\\_2011br.pdf](http://bstqb.org.br/uploads/syllabus/syllabus_ctfl_2011br.pdf)