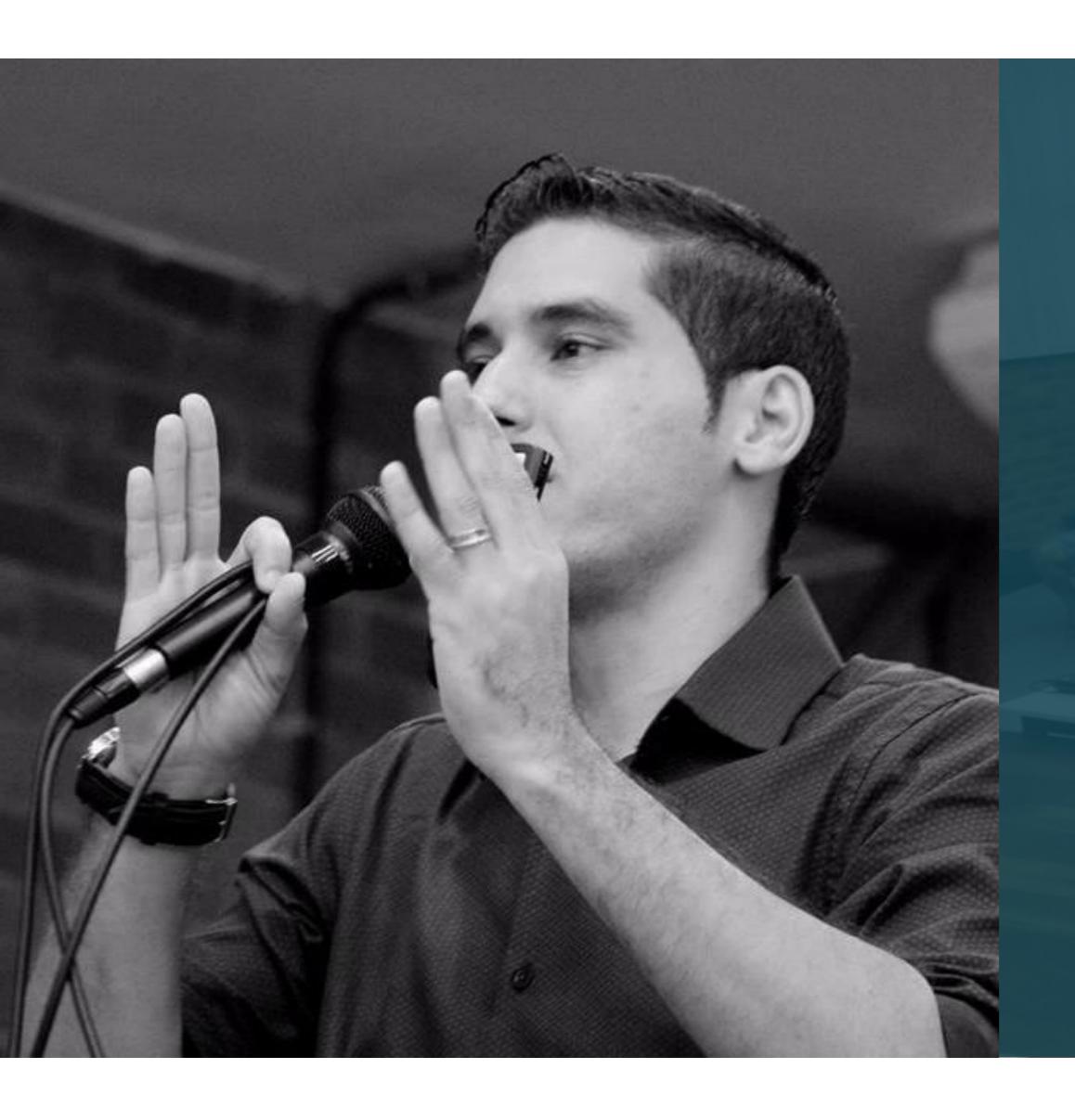
Teste de Software

Gustavo Farias

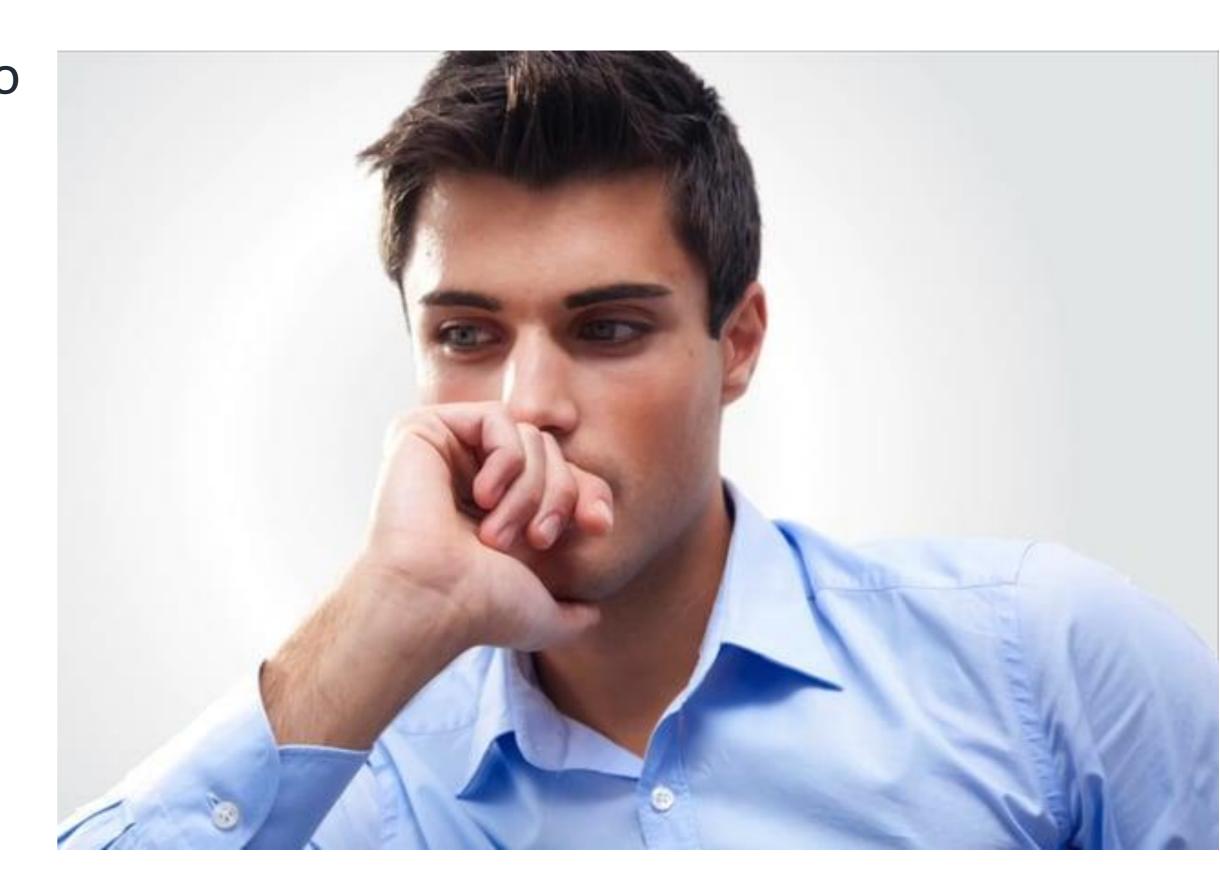
Quem sou eu



- 1. Formado em Ciência da Computação
- 2. Analista de Sistemas (Sun Certified Java Programmer)
- 3. Especialista em Teste de Software (BSTQB Certified)
- 4. Analista na Accenture por 6 anos
- 5. Especialista em Gestão de Projetos (PMP Certified)
- 6. Experiência em Outras Áreas do Mercado
- 7. Professor em Pós-Grad. em Diversas Universidades
- 8. Palestrante
- 9. Escritor
- 10. Coaching de Carreira
- 11. Empreendedor
- 12. Carreira Internacional

O que você acha que alcançará neste curso

- Dominar os princípios chaves e principais técnicas do teste de software
- Estar apto a assumir novas responsabilidades como um testador especialista
- Ter mais uma ferramenta para sua "caixa de ferramentas"
- Conseguir mais um certificado
- Melhorar o curriculo



O que você realmente alcançará



- 1. Tecnicamente:
 - . Todos os itens anteriores (a parte técnica)
- 2. Não técnico (mais importante):
 - . Orientações específicas sobre como você poderá utilizar o conhecimento e habilidades adquiridas neste curso para:
 - Alavancar a sua carreira profissional
 - Criar excelentes oportunidades profissionais para o resto da sua vida
 - Ser peça chave em projetos importantes

Agenda

- 1. O que é teste de software
- 2. A situação das empresas
- 3. O perfil dos profissionais
- 4. A primeira oportunidade
- 5. A visão, o Perfil e o Papel do Testador
- 6. Fundamentos do teste de software
 - a. Níveis de Teste (quando testar?)
 - b. Técnicas de Teste (como testar?)
 - c. Tipos de Teste (o que testar?)

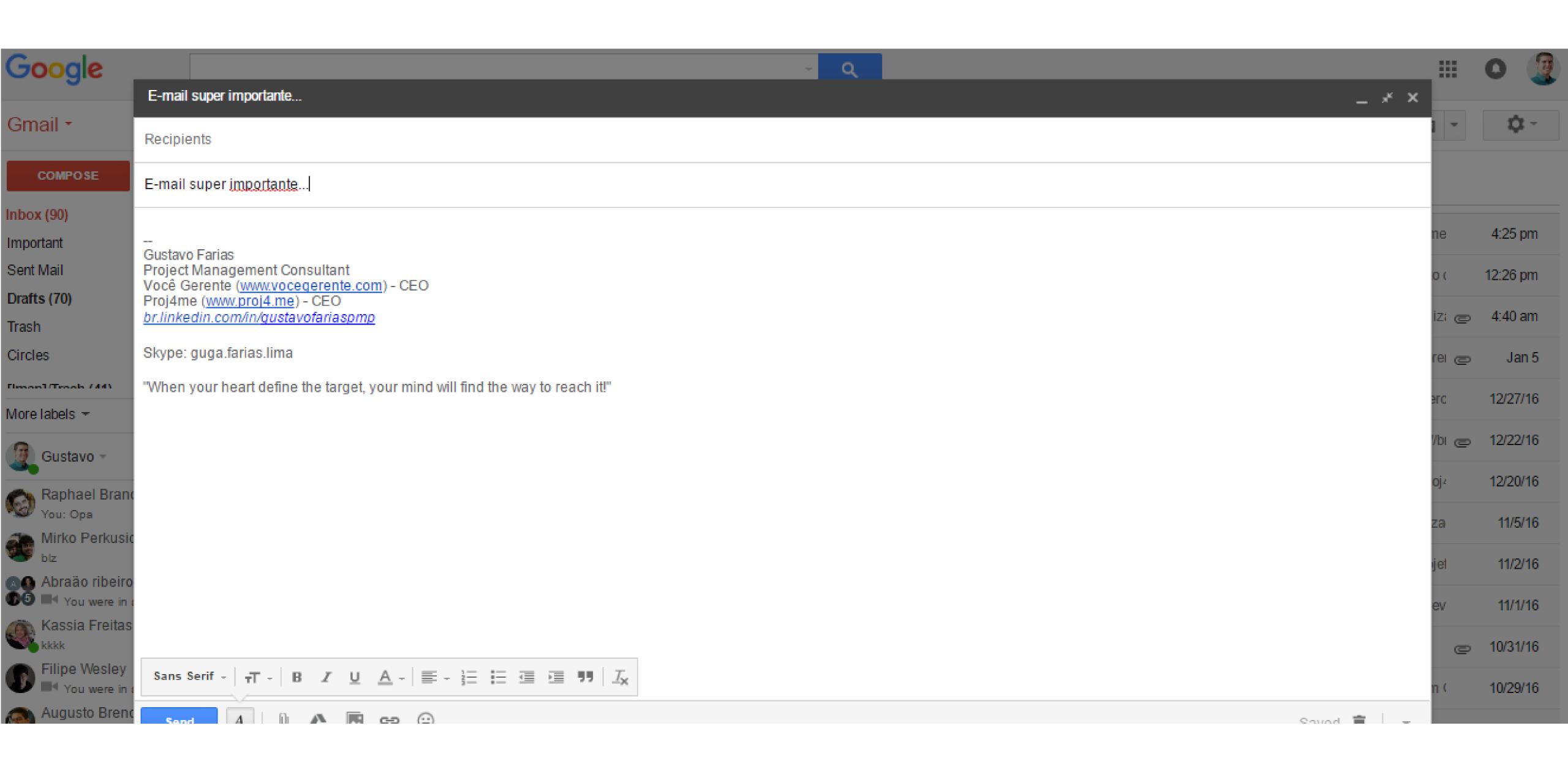
Agenda

- 7. Quando e como os testes entram no processo de desenvolvimento de software
- 8. Criação do Plano de Teste
 - a. Planilha bônus
 - b. Padrões e boas práticas
- 9. Executando o Plano de Teste
 - b. Padrões e boas práticas
- 10. Testes Manuais x Testes Automatizados
- 11. Certificação BSTQB
- 12. A segunda oportunidade: como transformar o aprendizado deste curso em grandes oportunidades para sempre

O que é Teste de Software?

"É a verificação feita sobre um sistema ou parte dele para garantir que uma determinada entrada produza, sempre, uma saída esperada."







Algumas verdades importantes Sobre o Teste de Software

Algumas verdades importantes



1. Testes não verificam completamente as saídas de um sistema, pois as entradas são infinitas



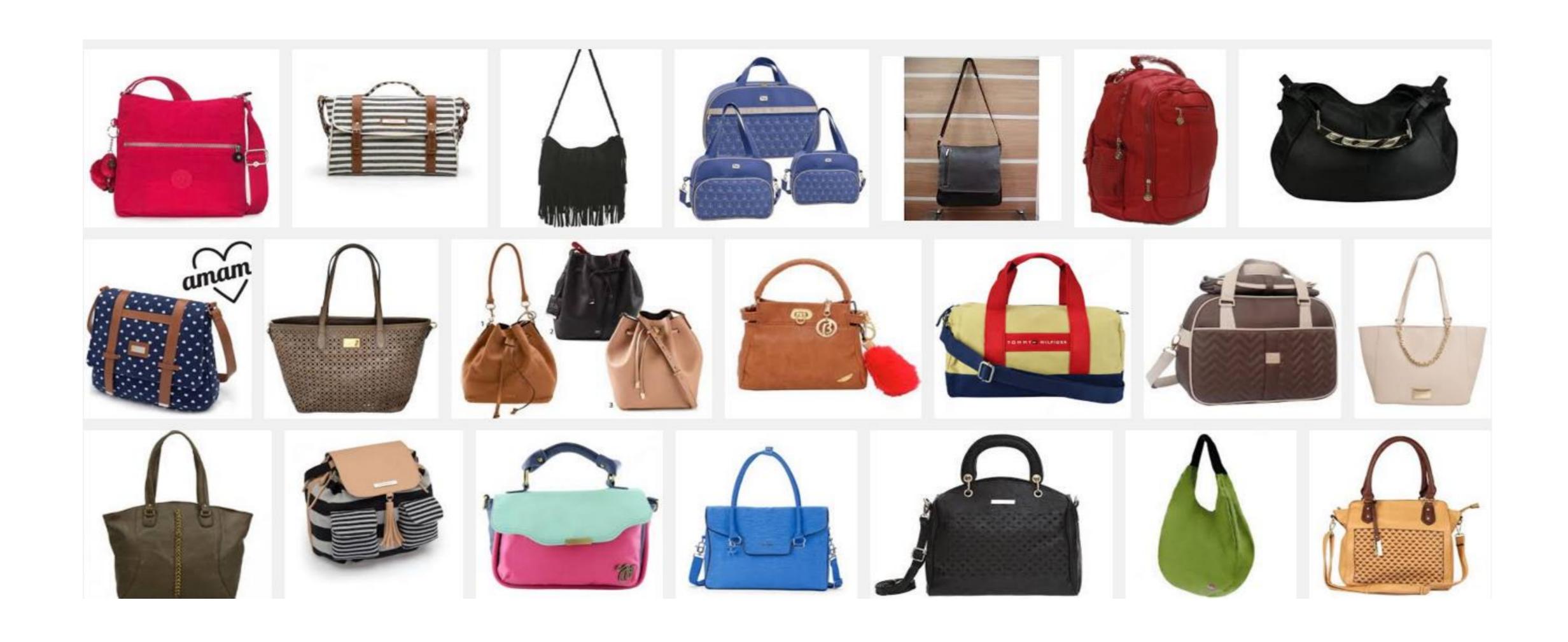


Algumas verdades importantes



- 1. Testes não verificam completamente as saídas de um sistema, pois as entradas são infinitas
- 2. Testes não garantem qualidade

Qualidade vai além de testes



Algumas verdades importantes



- 1. Testes não verificam completamente as saídas de um sistema, pois as entradas são infinitas
- 2. Testes não garantem a qualidade
- 3. Testes custam em média de 20% a 30% do custo total de elaboração de um software

"O teste não é essencial"



"O teste não é essencial..."

Disney: o jogo do Rei Leão



O primeiro jogo multimídia para crianças desenvolvido pela Disney teve uma das

maiores campanhas de maria de época.

As vendas ultrapar un teste de época.

No dia Faltou plataformativas.

No dia Faltou plataformativas.

No dia Faltou plataformativas.

As vendas ultrapar un teste de época.

No dia Faltou plataformativas.

As vendas ultrapar un teste de época. problemas ocorridos em algumas plataformas de PCs existentes no mercado.

Ariane 5



- Projeto da Agência Espacial Europeia custou
 - . 10 anos.
 - . US\$ 8 Bilhões.
- Vôo inaugural em 1996
- 40 segundos após a decolagem o foguete e sua carga avaliada em US\$ 500 milhões explodiram

Ariane 5



• Fatos:

- O foguete perdeu o controle de altitude e ativou a autodestruição
- Motivo: um programa que de vertia um número real para un teste de teiro de 16 bits recebeu um teste de valor fora da faixa realtou es limites valor fora da valores limites valor fora da valo
- Ocorrel erro de *out of range* que desnorteou os sistemas

Em 1983, uma III guerra mundial quase teve início devido a um BUG no software de alerta soviético. A falha fez com que o software mostrasse que os E.U.A. tinham lançado 5 mísseis.

Por sorte, o Tenente Coronel Stanislav Petrov considerou que a informação estava errada e não ordenou o contra-ataque, evitando assim o início de uma nova guerra.



A situação das empresas

A situação das empresas

- Prazos apertados para entrega de sistemas
- Clientes menos tolerantes a atrasos nas entregas
- Precisam lançar software de boa qualidade
- Clientes menos tolerantes a falhas

Precisam construir software melhores, mais rápidos e mais baratos.

A situação das empresas

- Novas tecnologias e sistemas cada vez mais complexos
- O mercado tem pouquíssimos testadores especialistas. Todos querem ser programadores!
- Não investem o suficiente em testes
- Software testados apenas pelos próprios desenvolvedores ("sistemas perfeitos")

Sistemas de qualidade abaixo do desejado, "mas é assim mesmo, vamos consertando depois de colocar em produção".

O perfil dos profissionais

O perfil dos profissionais

- O mercado tem pouquíssimos testadores especialistas. Todos querem ser programadores!
- Dos testadores, a maioria não são realmente especialistas, apenas foram inseridos em alguma equipe de teste da empresa
- Existe um pouco de preconceito com testes, que é falado como uma "parte chata do desenvolvimento de software"
- Encontrar um profissional realmente com o perfil, conhecimentos e habilidades de um testador é algo realmente raro



A primeira oportunidade

A primeira oportunidade

- O mercado tem pouquíssimos testadores especialistas. Todos querem ser programadores!
- Dos testadores, praticamente todos não são realmente especialistas, apenas foram transferidos para alguma equipe de teste
- Existe um pouco de preconceito com testes, que é falado como uma "parte chata do desenvolvimento de software"
- Encontrar um profissional especialista, com o perfil, conhecimentos e habilidades de um testador é algo realmente raro

Eu chamo isso de Oportunidade...

- Você é reconhecido e remunerado:
 - Pelo tamanho do problema que você resolve
 - Pela sua raridade
- Sim, existe uma carência no mercado e você vai aproveitar esta oportunidade!

A visão, o Perfil e o Papel do Testador

Visão Crítica

Visão voltada para a Qualidade

Detalhista

Criterioso

Paciente

Pessimista

Perspicaz

Focado

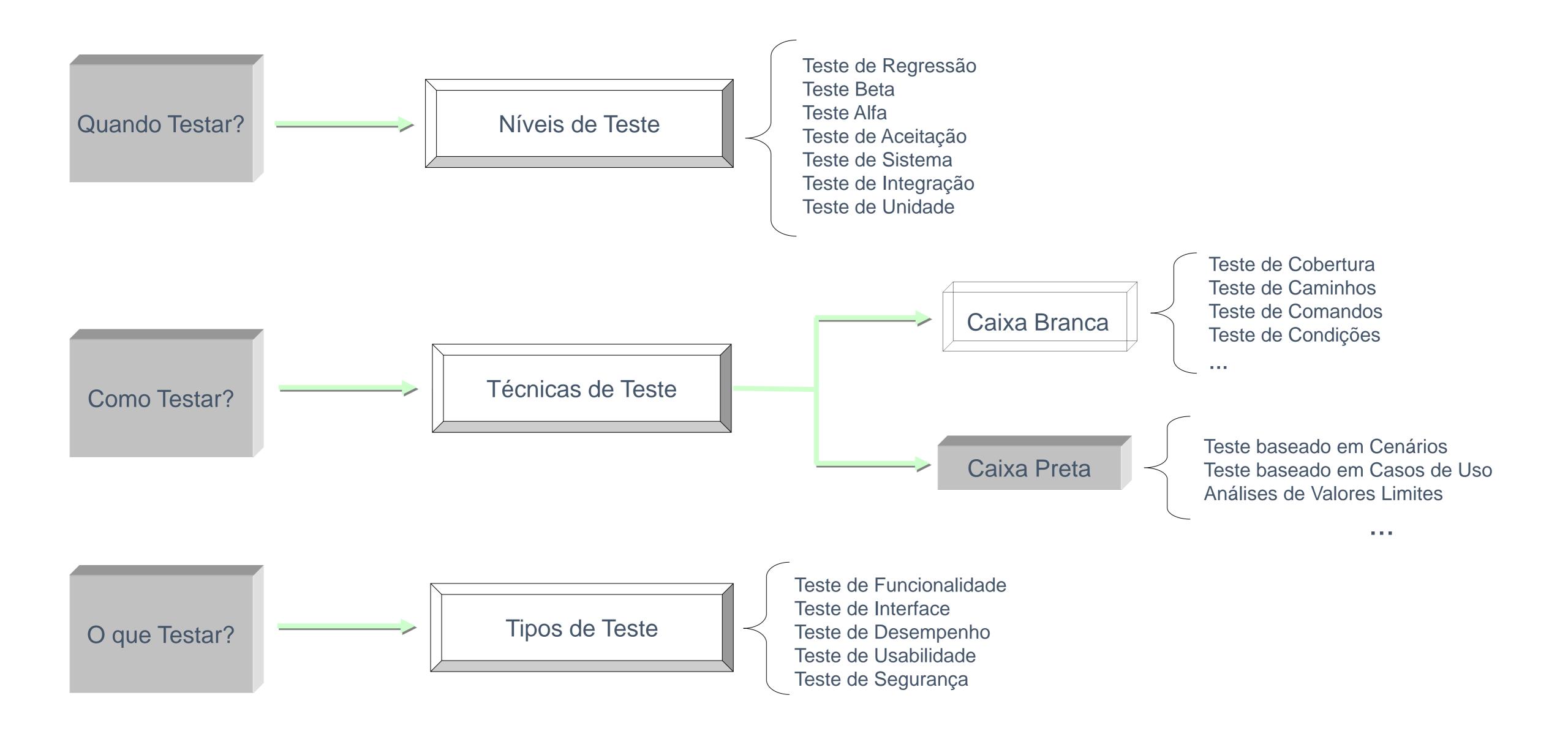
Motivado

Elaborar testes relevantes e exequíveis

Executar os testes no tempo planejado

Ser referência de critério e qualidade

Fundamentos do Teste de Software



Níveis de Teste

Quando Testar?

Níveis de Teste

Teste de Regressão

Teste Beta

Teste Alfa

Teste de Aceitação

Teste de Sistema

Teste de Integração

Teste de Unidade

Níveis de Teste

Quando Testar? Níveis de Teste

Teste de Regressão

Teste Beta

Teste Alfa

Teste de Aceitação

Teste de Sistema

Teste de Integração

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, i.e., 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

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

```
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
   st /
  public void testObterNomeCompleto() {
    Pessoa p = new Pessoa("Fulano", "Tal");
    assertEquals("Fulano Tal", p.getNomeCompleto());
```

Níveis de Teste

Quando Testar?

Níveis de Teste

Teste de Regressão

Teste Beta

Teste Alfa

Teste de Aceitação

Teste de Sistema

Teste de Integração

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
 - Ex.: 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

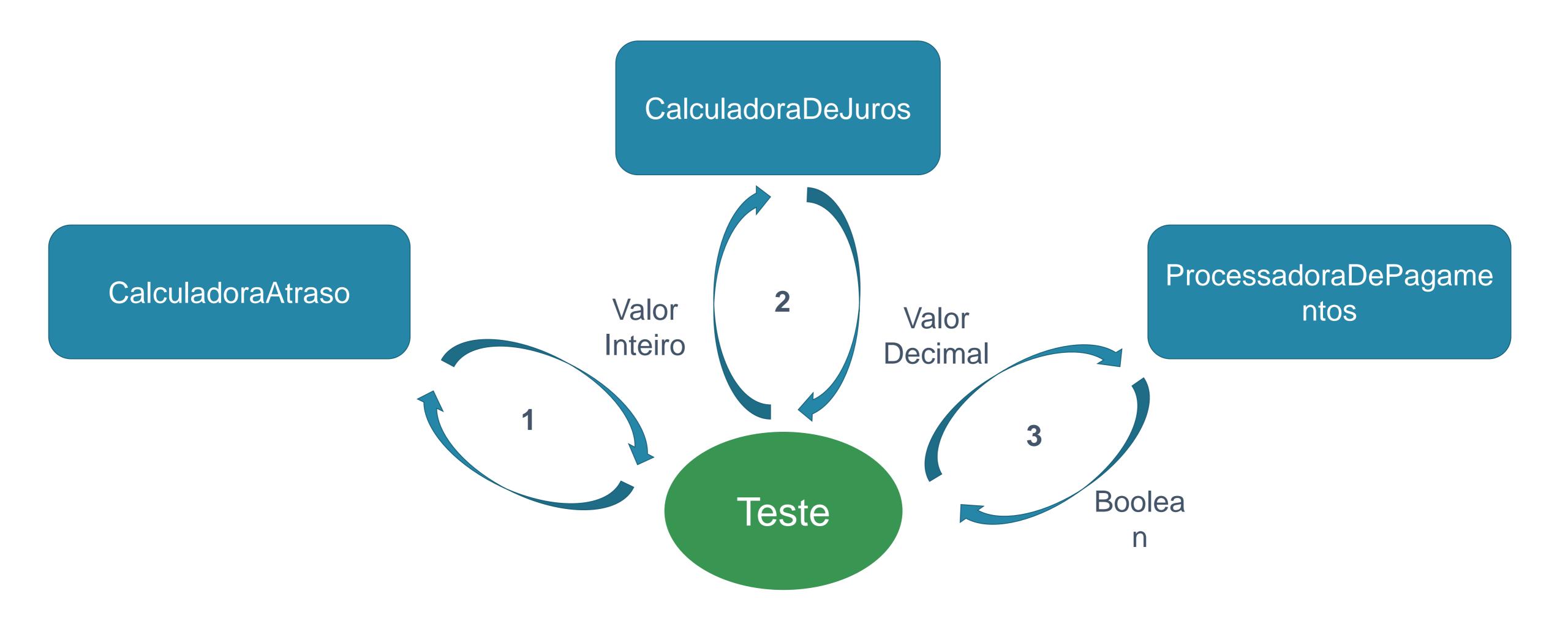
Classe 01



Classe N

Banco de Dados

Teste de Integração



Níveis de Teste

Quando Testar?

Níveis de Teste

Teste de Regressão

Teste Beta

Teste Alfa

Teste de Aceitação

Teste de Sistema

Teste de Integração

Teste de Sistema

- **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



Teste de Sistema







Níveis de Teste

Quando Testar?

Níveis de Teste

Teste de Regressão

Teste Beta

Teste Alfa

Teste de Aceitação

Teste de Sistema

Teste de Integração

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



Banco de Dados









Níveis de Teste

Quando Testar?

Níveis de Teste

Teste de Regressão

Teste Beta

Teste Alfa

Teste de Aceitação

Teste de Sistema

Teste de Integração

Teste Alfa

- **Objetivo:** executar o sistema de forma não planejada, sob ponto de vista de seu usuário final, porém, apenas por um grupo <u>pequeno</u> 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

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

Quando Testar?

Níveis de Teste

Teste de Regressão

Teste Beta

Teste Alfa

Teste de Aceitação

Teste de Sistema

Teste de Integração

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
 - Ex.: 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

Interface Gráfica (ex.: Browser)

Componente de Negócio 01 Componente de Negócio 02

Componente de Negócio 03

Classe 01



Classe N

Classe 01



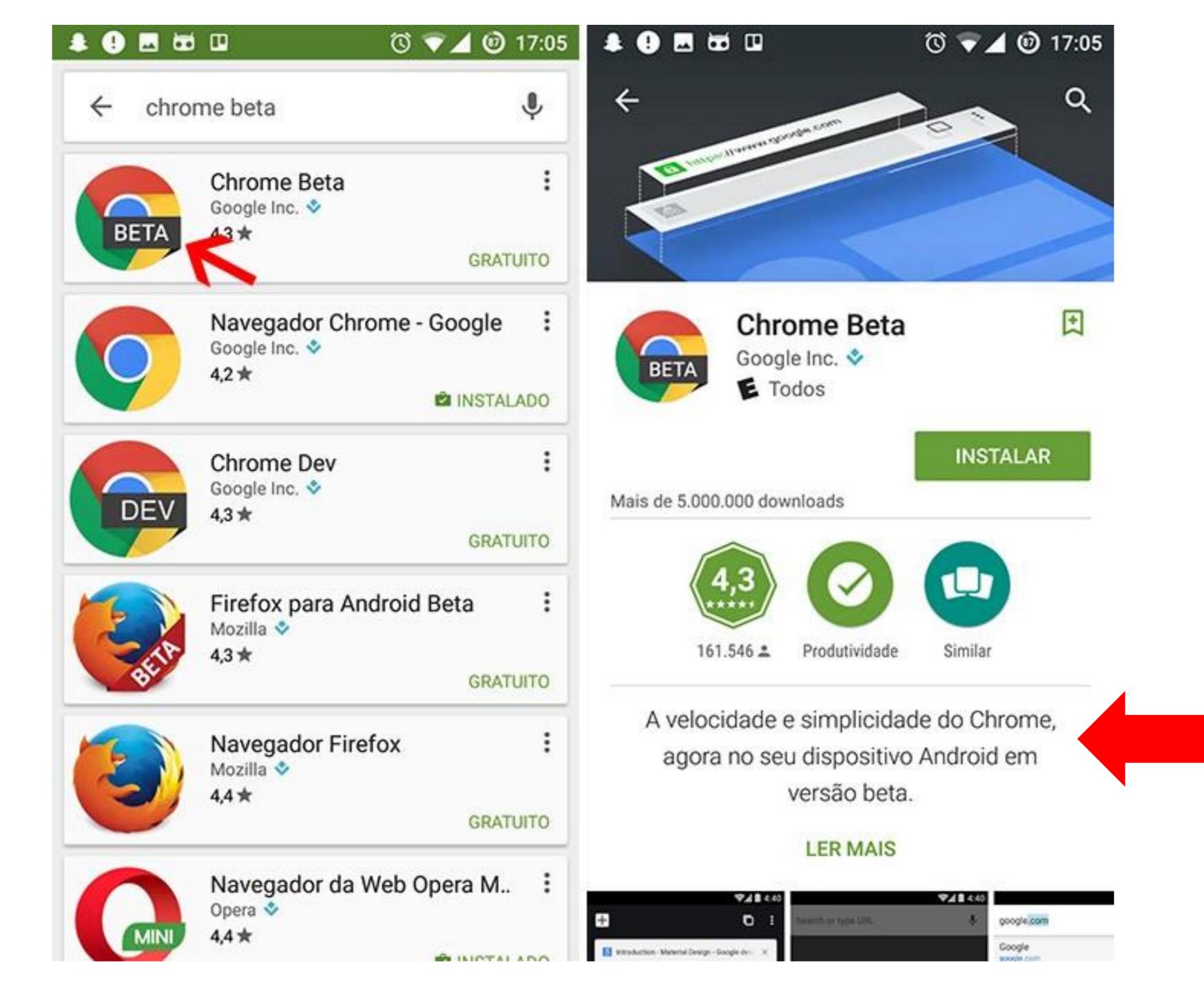
Classe N

Classe 01



Classe N

Banco de Dados



Níveis de Teste

Quando Testar?

Níveis de Teste

Teste de Regressão

Teste Beta

Teste Alfa

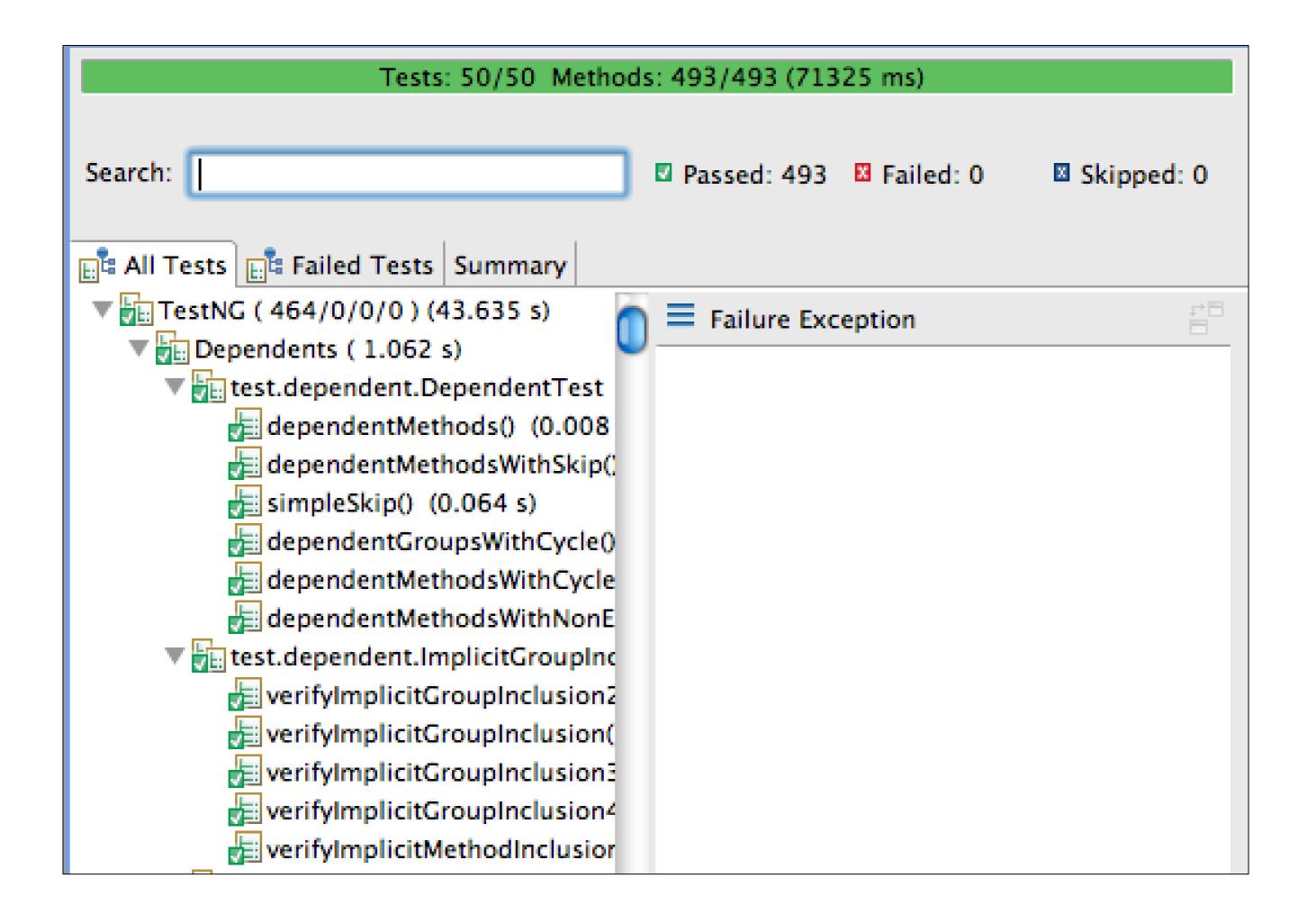
Teste de Aceitação

Teste de Sistema

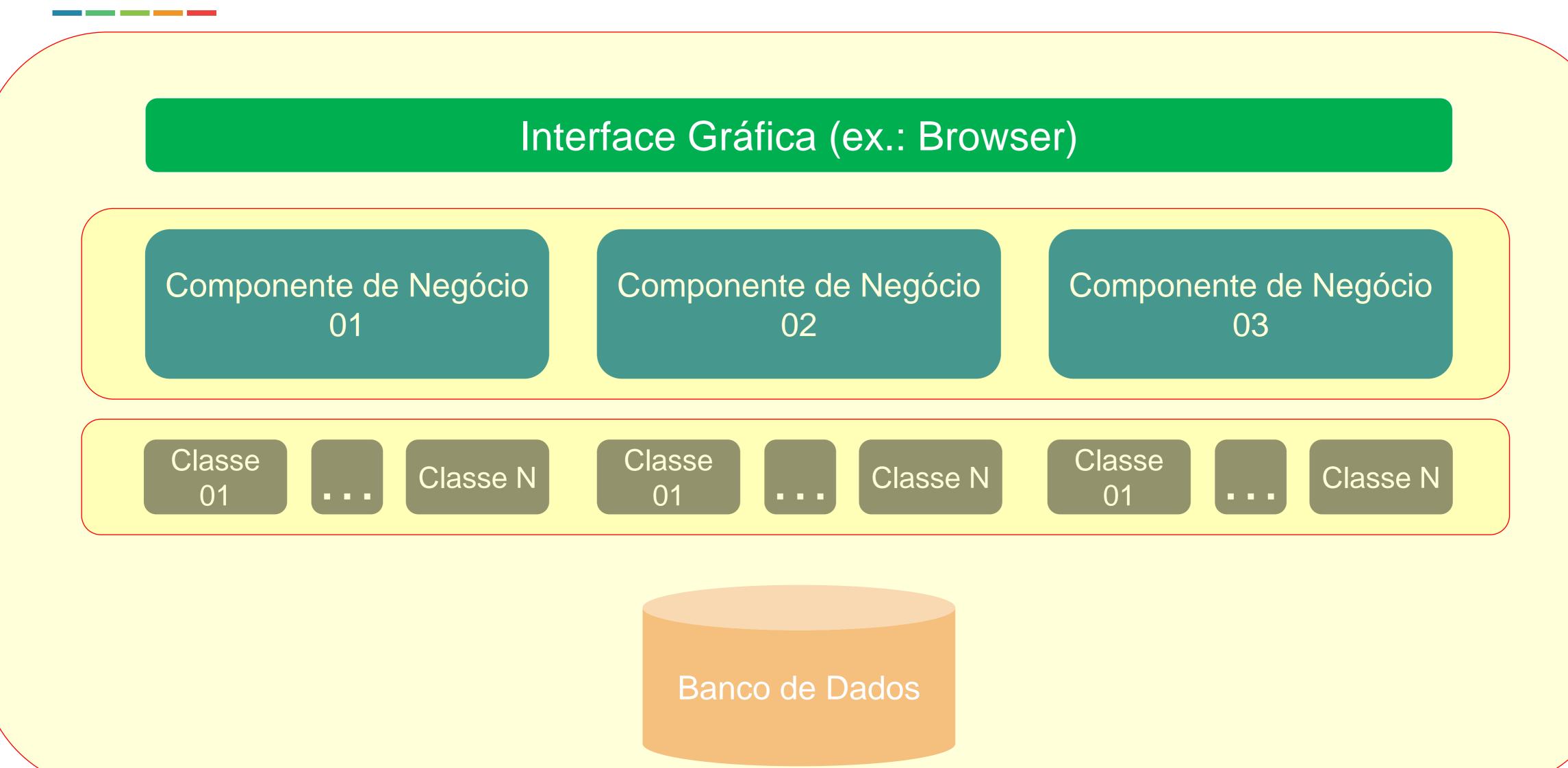
Teste de Integração

Teste Regressão

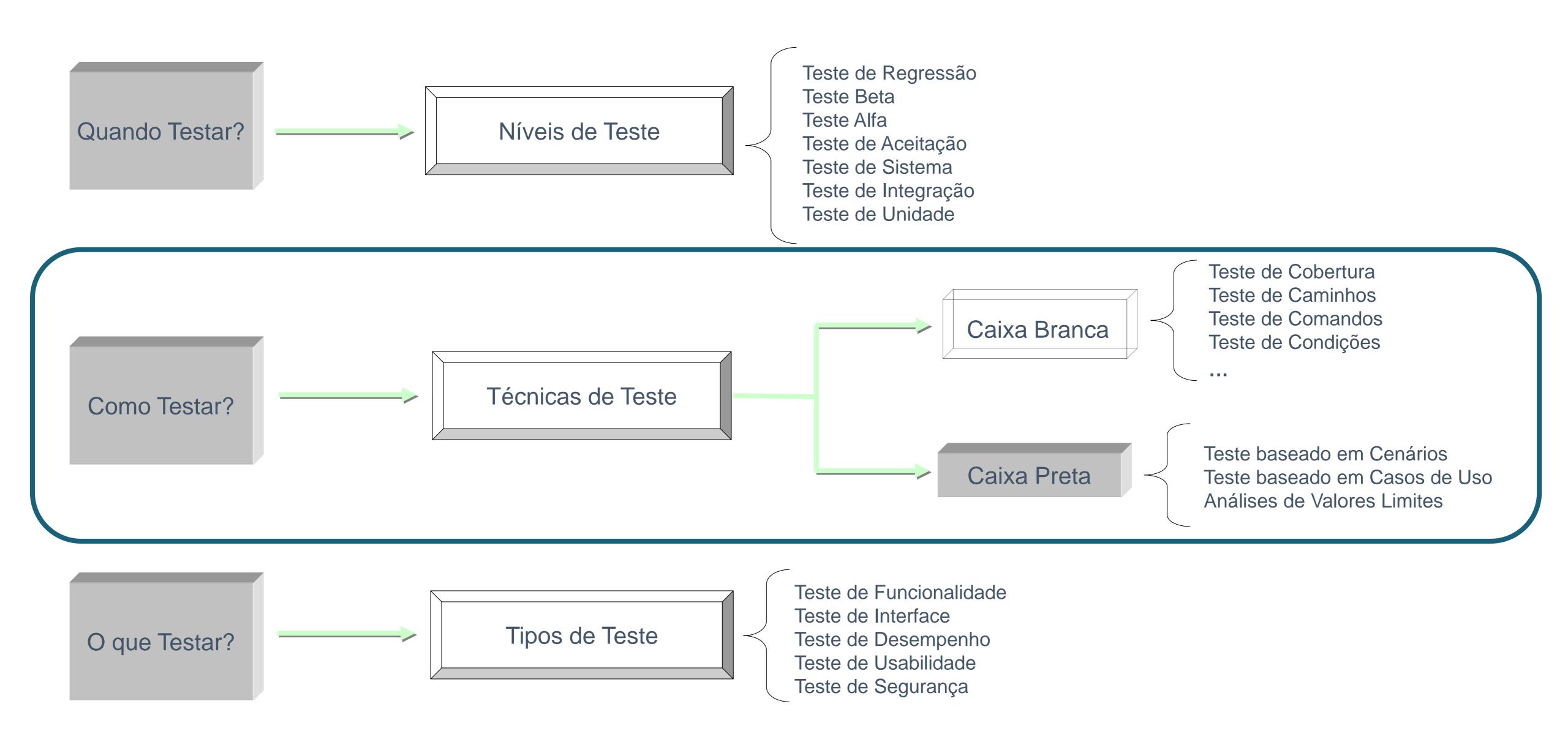
- **Objetivo:** reexecutar testes após alterações serem realizadas no sistema, para conferir se tudo continua funcionando corretamente (detectar efeitos colaterais):
 - Após mudanças de regras durante o desenvolvimento
 - Ex.: facebook agora permite a postagem de vídeos 3D na timeline
 - Após a correção 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



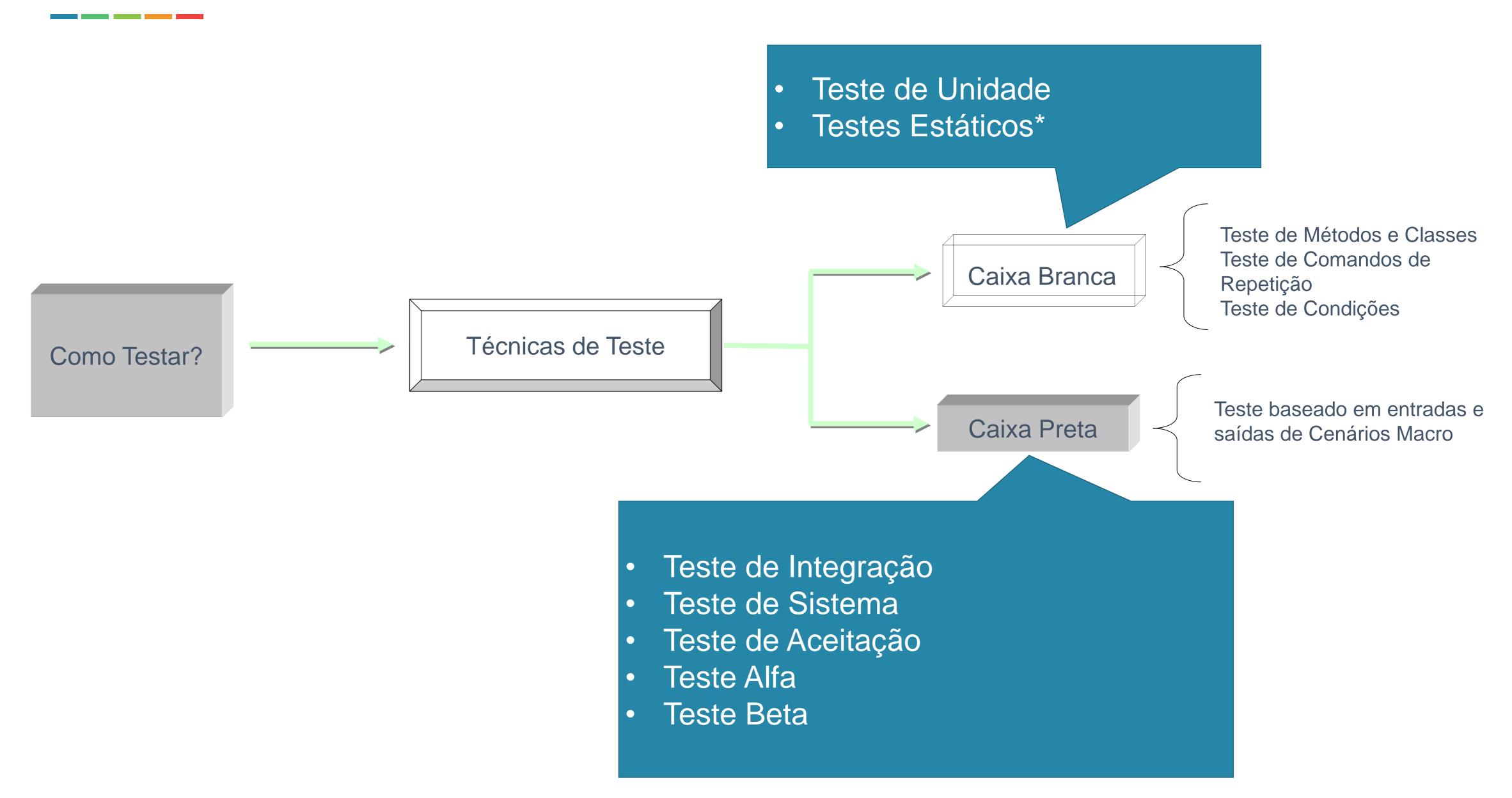
Teste de Regressão



Técnicas de Teste



Técnicas de Teste



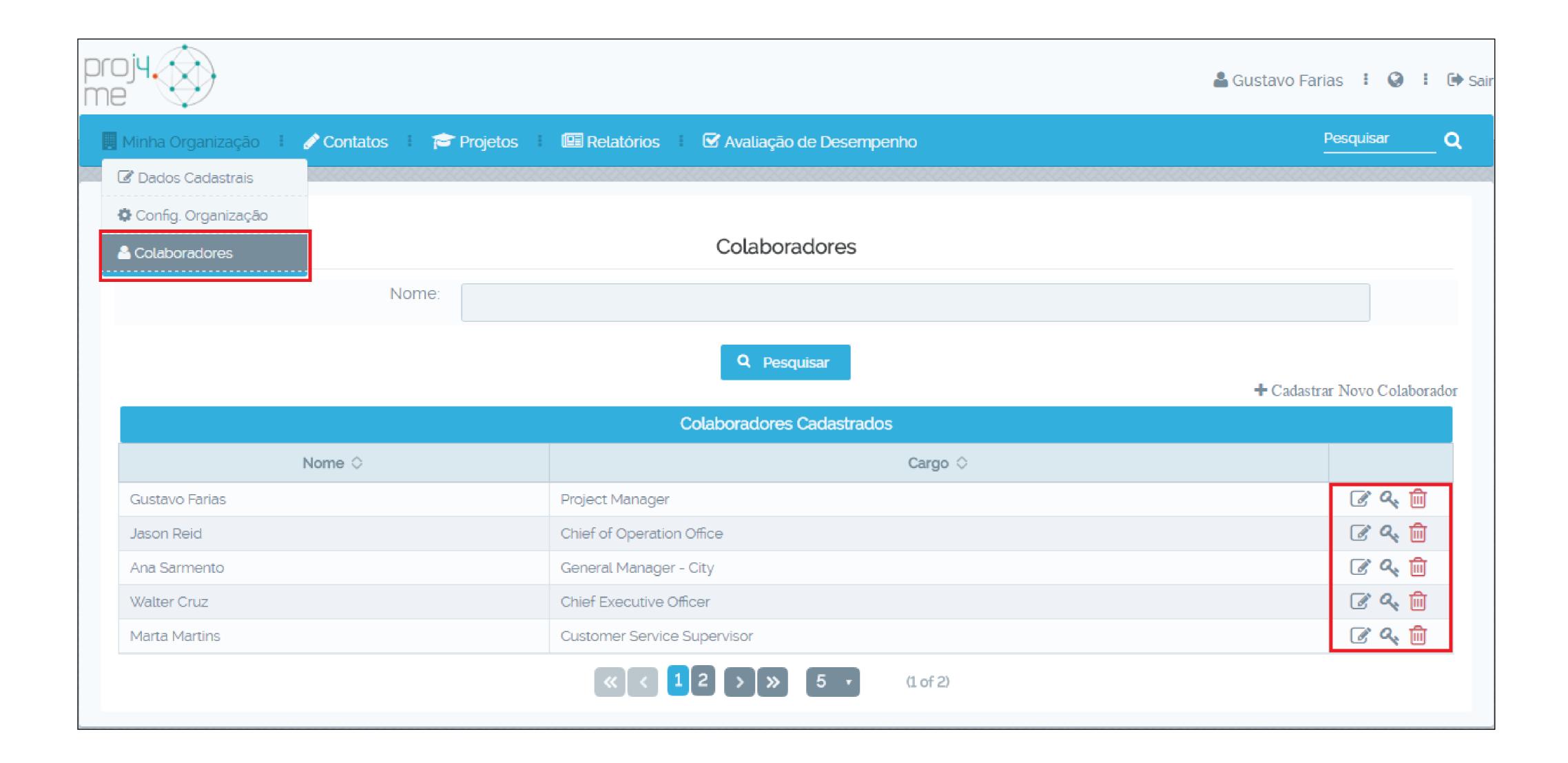
Teste Caixa Branca

```
1 package login;
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.ResultSet;
6 import java.sql.Statement;
8 public class User
     public Connection conectarBD() {
          Connection conn = null;
          try{
              Class.forName("com.mysql.Driver.Manager").newInstance();
              String url = "jdbc:mysql://127.0.0.1/test?user=lopes&password=123";
              conn = DriverManager.getConnection(url);
          }catch (Exception e) { }
          return conn; }
     public String nome="";
      public boolean result = false;
      public boolean verificarUsuario (String login, String senha) {
          String sql = "";
          Connection conn = conectarBD();
          //INSTRUÇÃO SQL
          sql += "select nome from usuarios ";
          sql +="where login = " + "'" + login + "'";
          sql += " and senha = " + "'" + senha + "';";
          try{
              Statement st = conn.createStatement();
              ResultSet rs = st.executeQuery(sql);
              if(rs.next()){
                  result = true;
                  nome = rs.getString("nome");}
          }catch (Exception e) {
          return result; }
      }//fim da class
```

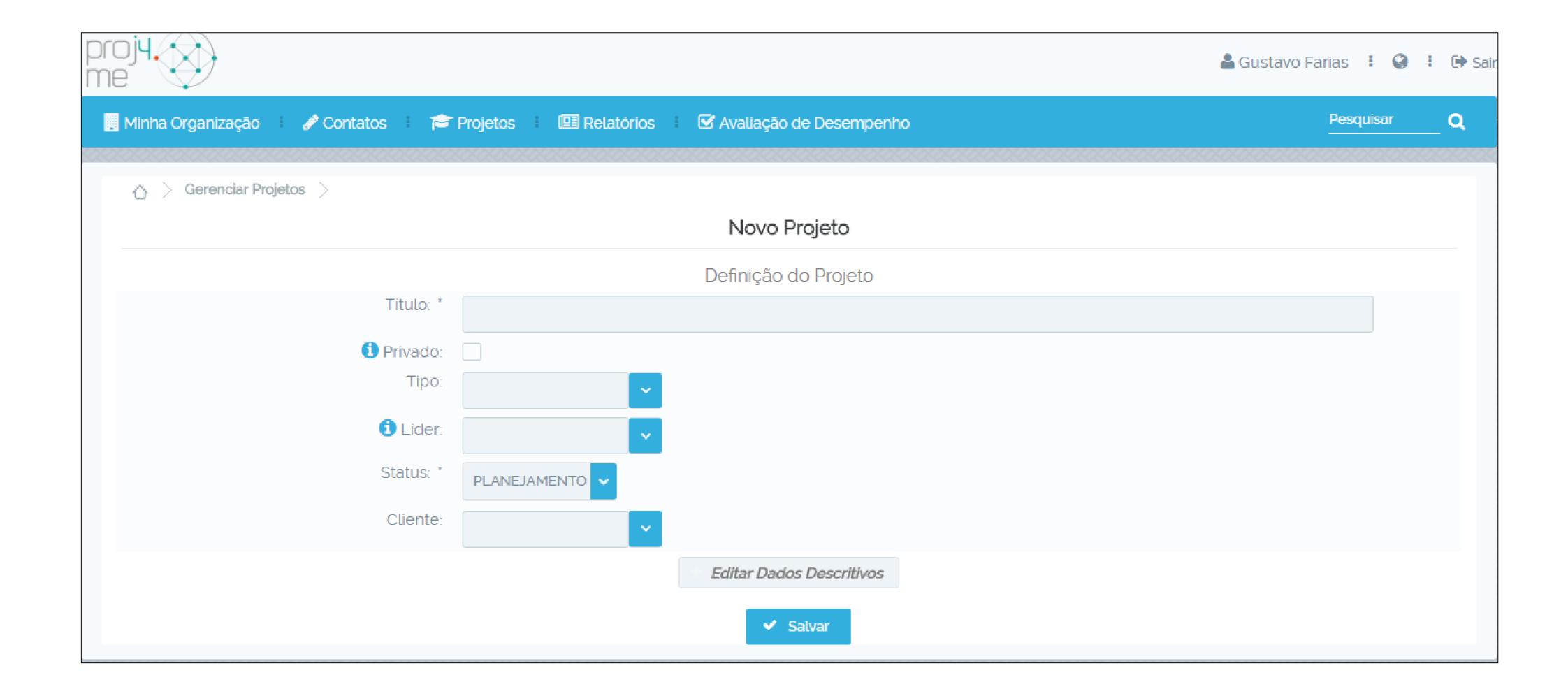
Análise Estática:

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

Teste Caixa Preta



Teste Caixa Preta



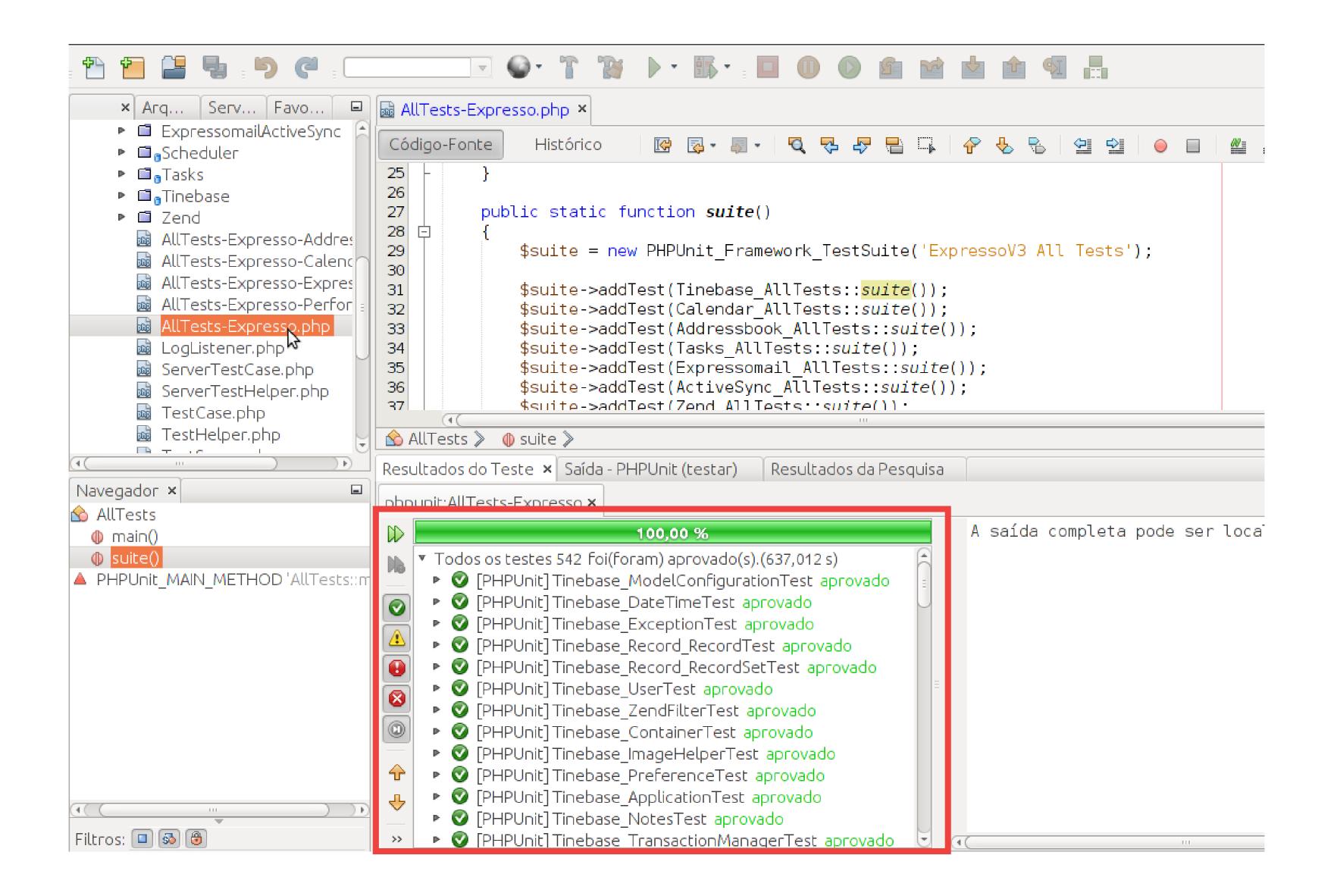
Testes Manuais x Testes Automáticos



Testes Automáticos

```
import junit.framework.TestCase;
class MyMathTest extends TestCase {
 public void testMdcPositive() {
    assertEquals(1, MyMath.mdc(2, 3));
    assertEquals (2, MyMath.mdc(2, 4));
 public void testMdcNegative() {
   assertEquals (2, MyMath.mdc(-4, 6));
 public void testMdcZero() {
    try { MyMath.mdc(0,1); fail(); }
    catch(IllegalArgumentException e){}
```

Testes Automáticos: a grande vantagem



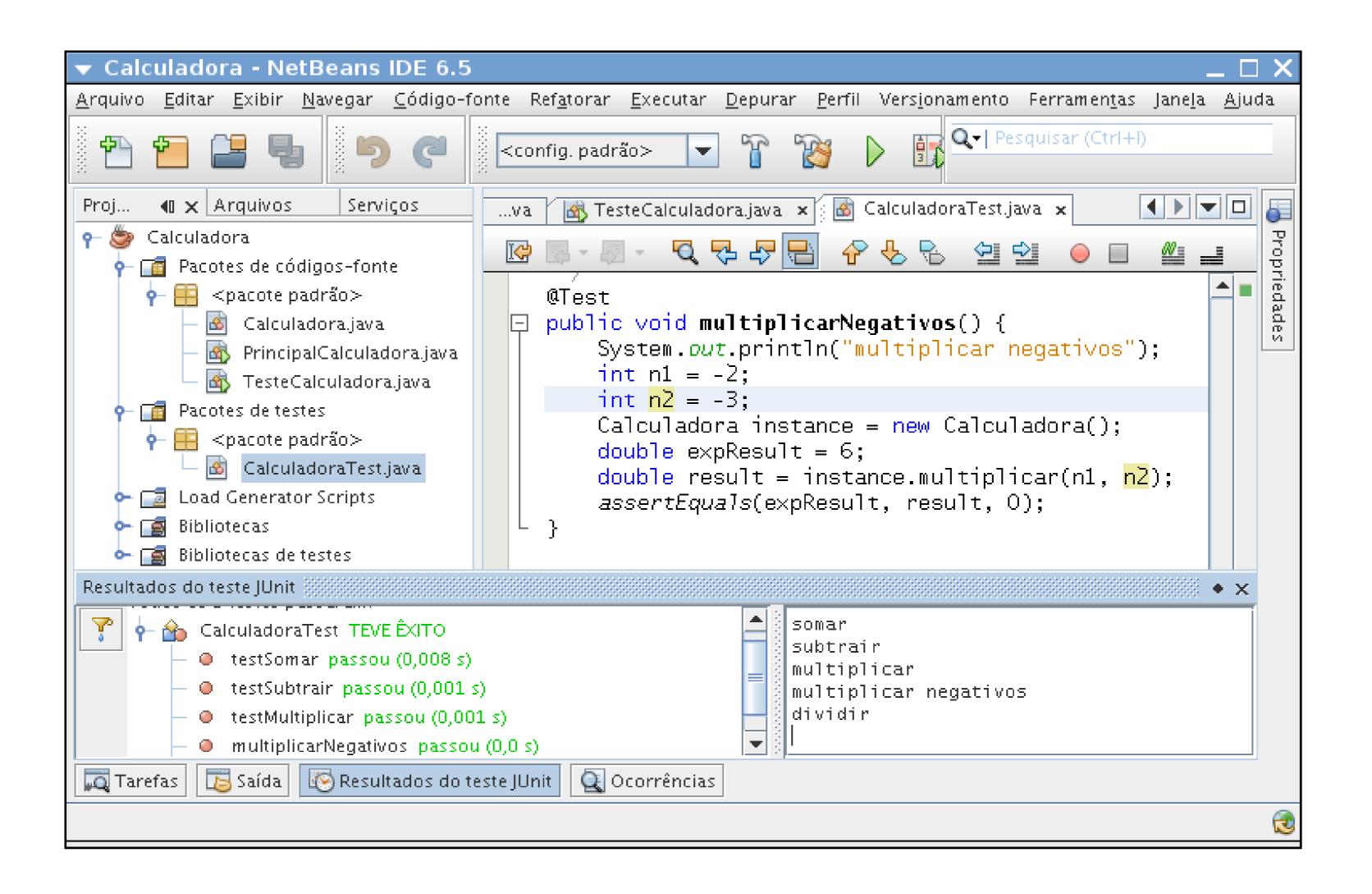
Testes Manuais	Testes Automáticos
Velocidade de execução baixa	Execução muito rápida
Repetitivo e Cansativo	Não cansam e não sentem preguiça.
Não exige tecnologias	Exige domínio de tecnologias específicas
Alto custo a cada execução	Alto custo apenas na criação
Possuem limitações quando o teste envolve situações de grande paralelismo	Permite testar situações impossíveis de testar manualmente
Podem explorar além do cenário de teste, quando necessário	Faz apenas o programado para fazer
Podem avaliar questões visuais como cores e formas	Não avaliam questões visuais
Podem avaliar questões de usabilidade	Não avaliam questões de usabilidade

Teste Estático x Teste Dinâmico

Teste Estático (Análise Estática)

- Objetivo: analisar o código sem executá-lo e verificar se as boas práticas adotadas foram obedecidas
- Obviamente é um teste "Caixa Branca"
- Existem ferramentas que fazem este teste automaticamente
 - Ex.: <u>Findbugs</u>
- Exemplos de verificações realizadas:
 - Código está documentado?
 - Variáveis e constantes possuem boa nomenclatura?
 - Código está organizado e com boa legibilidade?
 - Existem possíveis NullPointers não tratados pelo programador?
 - O programador obedeceu à arquitetura do sistema ao codificar?
 - As conexões com o banco de dados estão sendo fechadas após utilizadas?

Teste Estático (Análise Estática)



Teste Estático (Análise Estática)

Projeto / Sistema:
Revisor:

DD/MM/AAAA

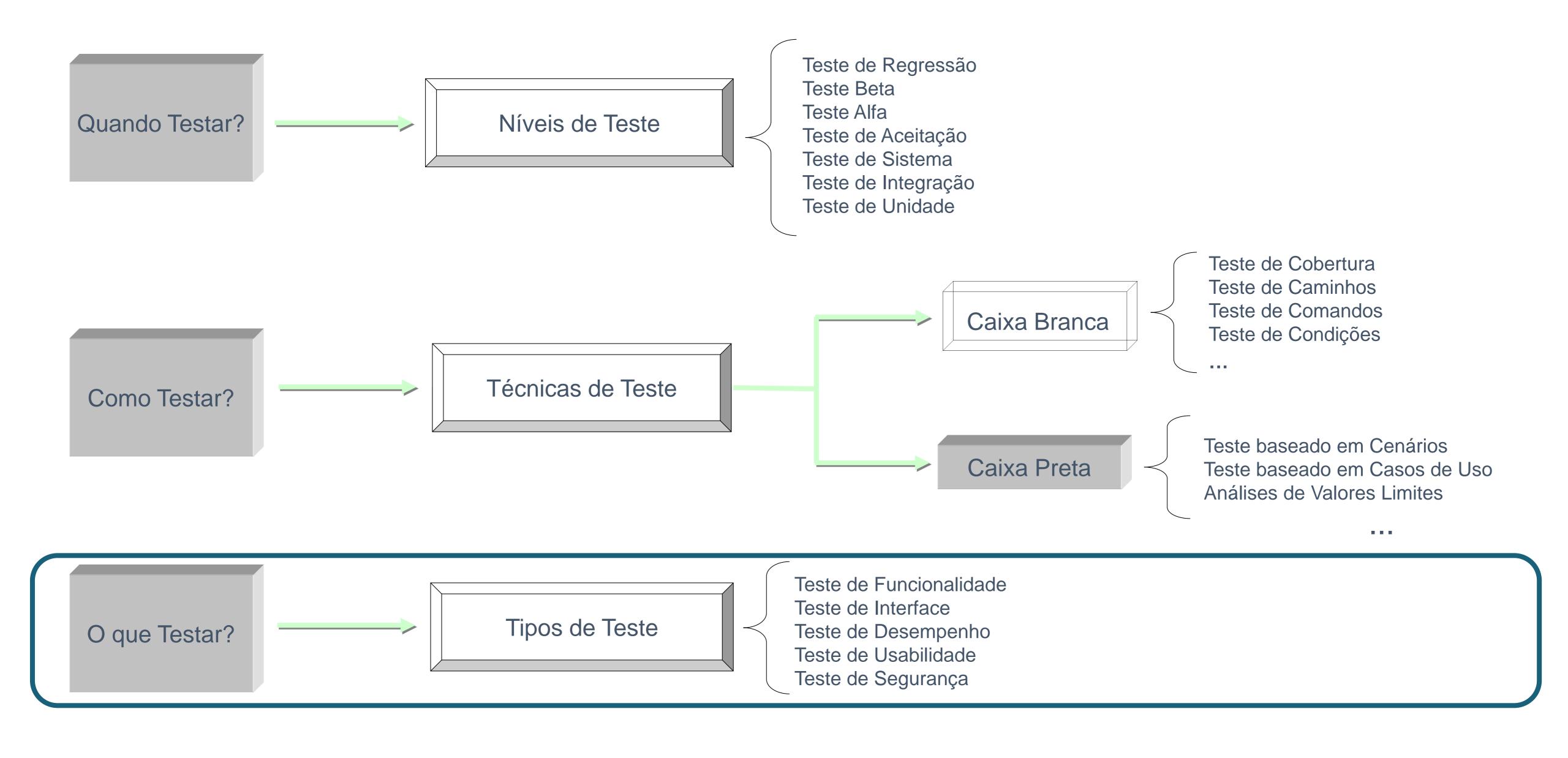
Data da Revisão:

#	ltem	Status	Artefato com Erro	Correções a Serem Realizadas
1	O código está documentado conforme os padrões da empresa?	Não	Método CalculadoraDeJuros.calculajuros();	Faltou documentar o método apontado.
2	Os arquivos e classes estão com nomenclatura significativa?	Sim		
3	As variáveis e constantes estão com nomenclatura significativa?	Sim		
4	Existe algum loop sem condição de parada?	Sim		
5	Existe algum ponto de NullPointer não tratado?	Sim		
6	Existe alguma variável não inicializada?	Não	Método CalculadoraDeJuros.calculajuros();	A variável "Double jurosInicial" não está sendo inicializada.
7	O código obedece à arquitetura pré-definida para o sistema?	Não	Método CalculadoraDeJuros.calculajuros();	A abertura da conexão com o BD deve acontecer na camada de negócio.
8	Foi realizado o tratamento de exceções?	Não	Método CalculadoraDeJuros.calculajuros();	Não está sendo realizado o tratamento de exceções.
9		Sim		
10		Sim		

Teste Dinâmico

- Objetivo: validar o sistema através de sua execução
- É o método tradicional: insere uma entrada, executa o sistema e confere a saída.
- As análises estáticas e dinâmicas se complementam

Tipos de Teste



Tipos de Teste

O que Testar?

Teste de Desempenho

Teste de Usabilidade

Teste de Segurança

Teste de Portabilidade

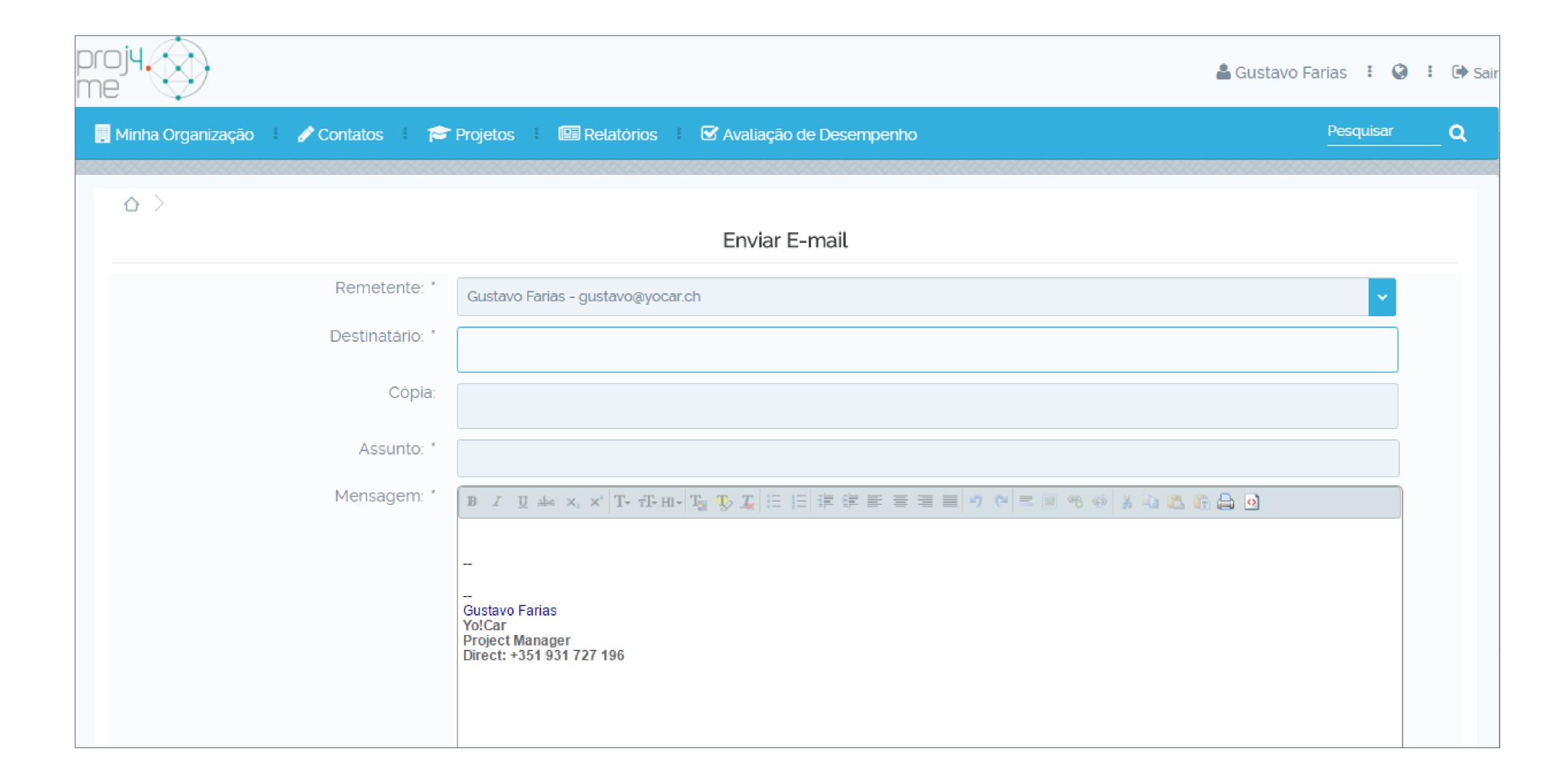
Teste de Funcionalidade

Teste de Stress

Teste de Funcionalidade

- Objetivo: validar que as funcionalidades do sistema estão funcionando corretamente
- Envolve:
 - Unidade
 - Integração
 - Sistema
 - Aceitação
 - Alfa
 - Beta
 - Regressão

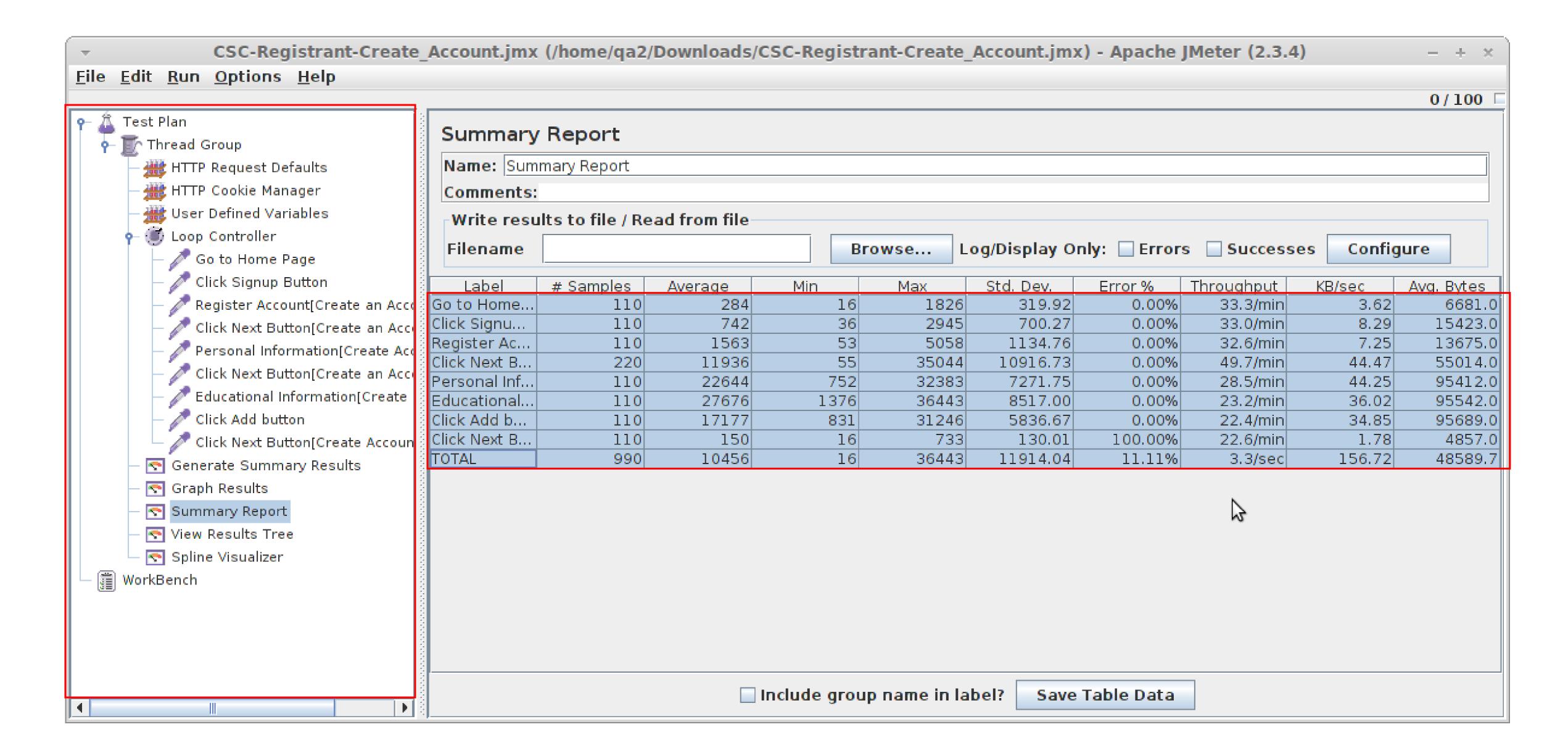
Teste de Funcionalidade



Teste de Desempenho

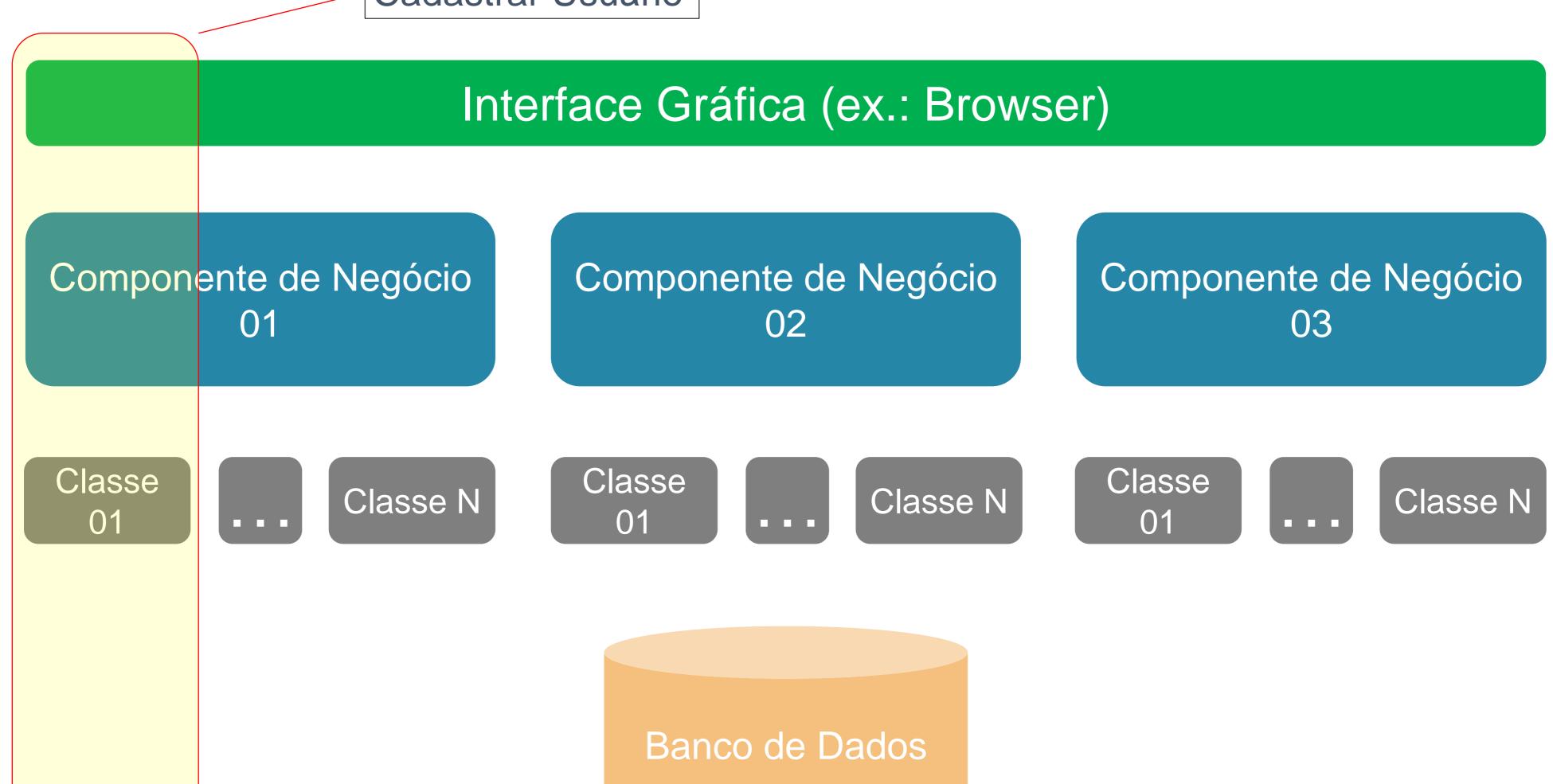
- Objetivo: validar o desempenho do sistema no que diz respeito ao seu tempo de resposta para determinadas operações
- Exemplos:
 - Tempo de resposta de um sistema bancário para processar uma operação
 - Tempo de resposta do facebook para fazer upload de fotos
 - Tempo de resposta de um robô médico para responder a comandos
- Ferramenta:
 - Jmeter

Teste Desempenho



Teste Desempenho

Cadastrar Usuário

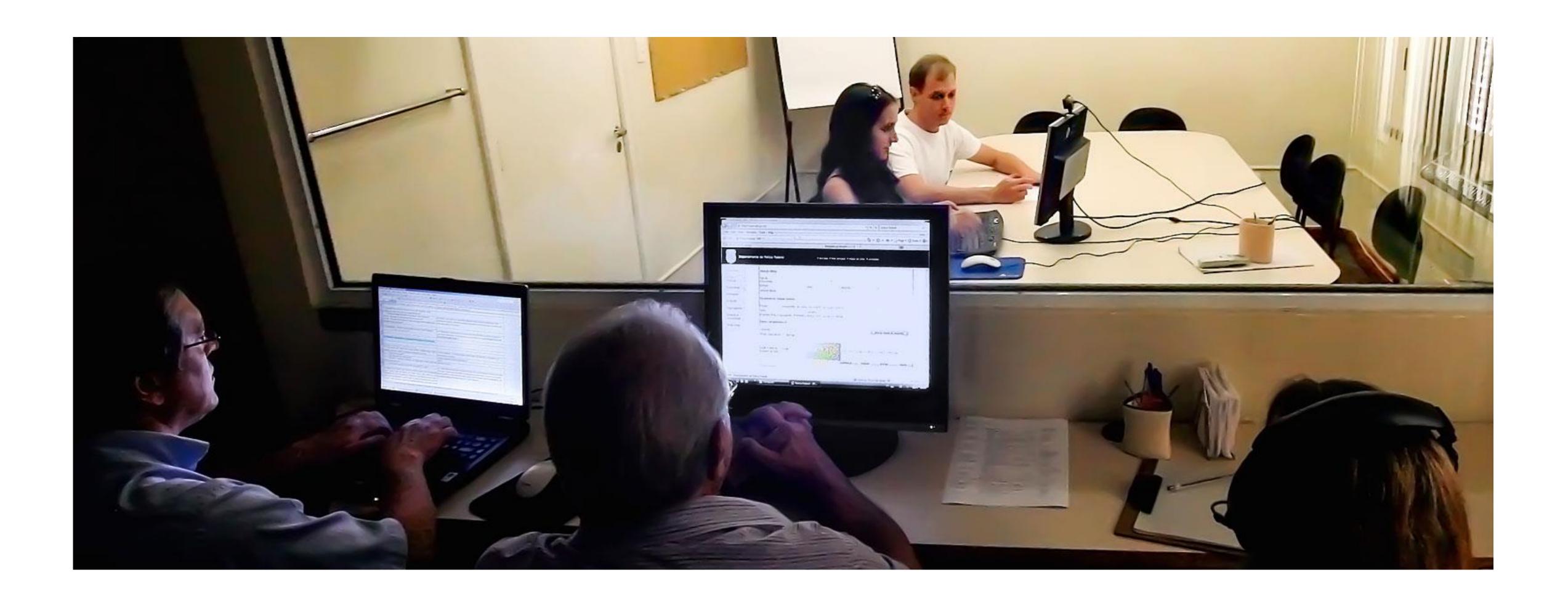


Teste de Usabilidade

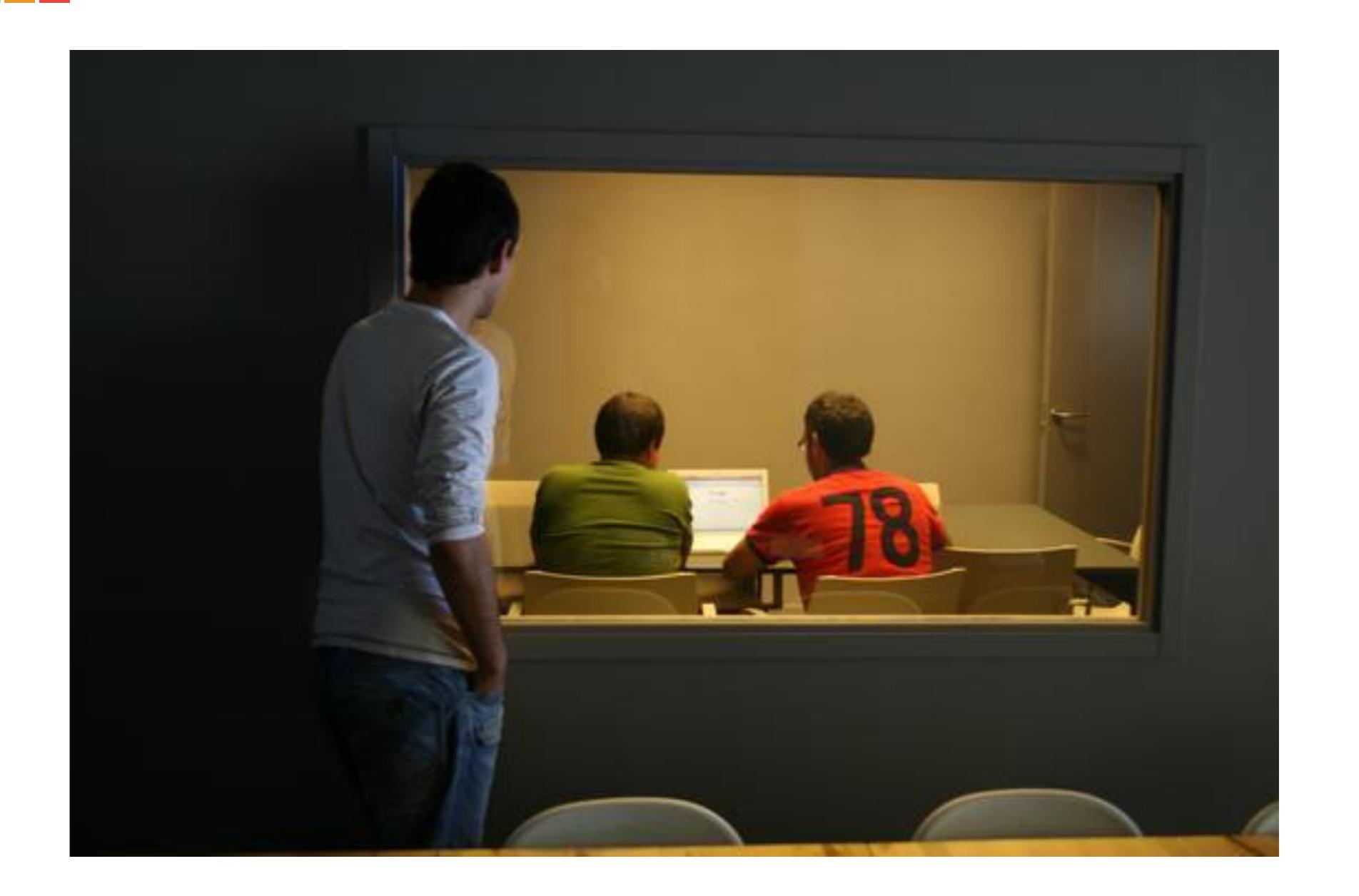
- Objetivo: validam aspectos que envolvem a experiência do usuário ao utilizar o sistema
- Exemplos:
 - Estética de um website
 - Definição e disposição de cores da interface do aplicativo
 - Tipo de interface a ser utilizada
 - Touch screen? Gestos? Sensível ao som?
 - Ajuda on-line e contextual
 - Manual do usuário



Teste de Usabilidade



Teste de Usabilidade



- **Objetivo:** validam a proteção do sistema contra invasões ou acesso não autorizado a informações
- Exemplos:
 - Sites com acesso restrito
 - Tráfego de informações criptografadas







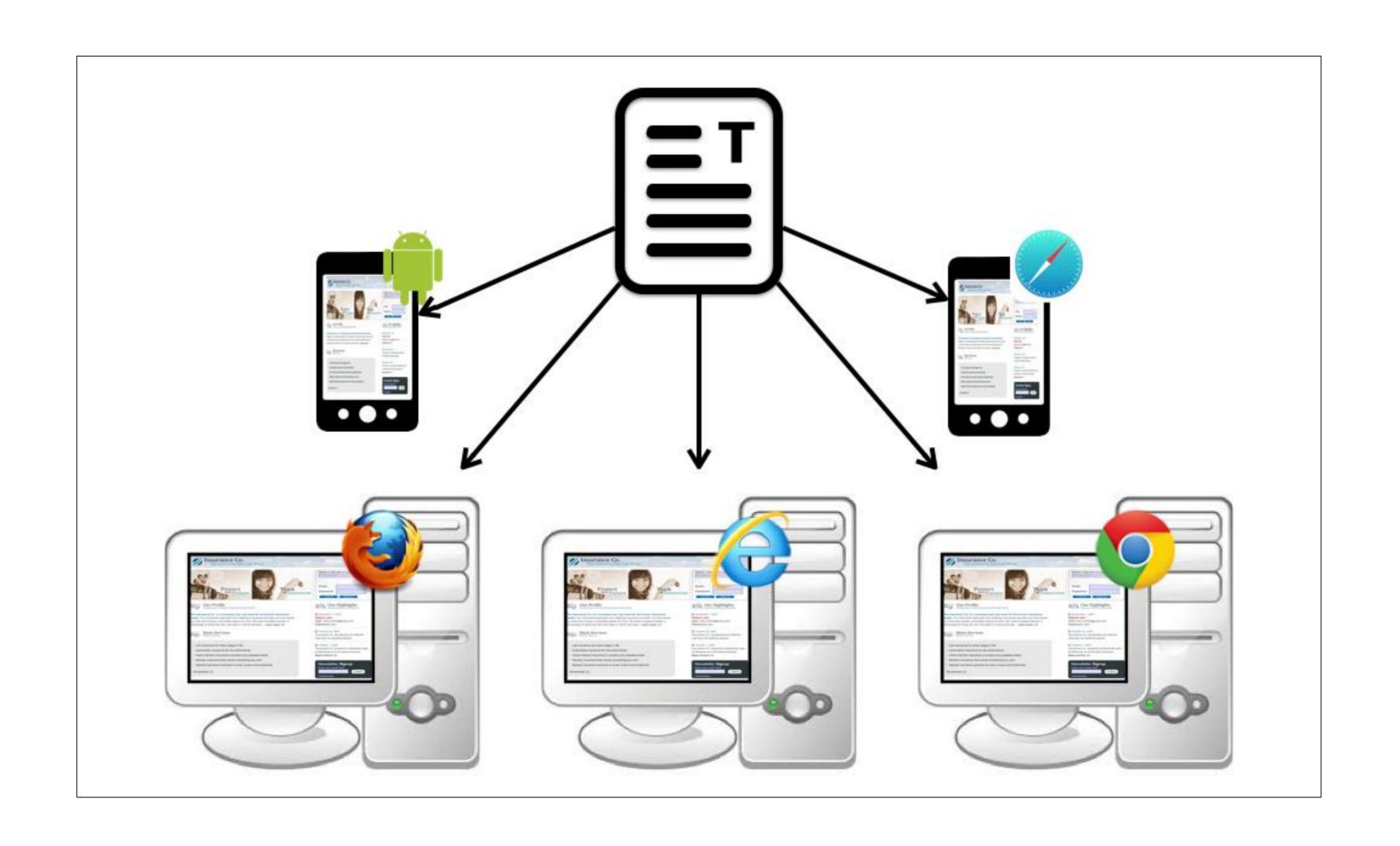
Teste de Portabilidade

- Objetivo: valida o funcionamento do sistema em diferentes plataformas e dispositivos, nas quais o sistema está proposto a funcionar
- Exemplos:
 - IOS, Android, etc.
 - Smarphones, Tablets, Notebooks, etc.

Teste de Portabilidade



Teste de Portabilidade



Teste de Stress

- Objetivo: validam o comportamento do sistema em condições extremas.
- O stress no sistema pode abranger cargas de trabalho extremas, memória insuficiente, hardware e serviços indisponíveis ou recursos compartilhados limitados
- Executados o quanto antes para se ter o tempo necessário para corrigir ou melhorar o desempenho do sistema conforme necessário
- Exemplos:
 - Sobrecarga de acesso a um website
 - Sobrecarga de processamento em um sistema de gestão de clientes
 - Ex.: extração de relatórios grandes em paralelo

confiabilidade fumaçamantenabilidade volume operacional interoperabilidade funcionalidade robustez USabilida manute manutenção compatibilidade es recuperabilidade concorrência

Quando e Como os Testes Entram no Processo de Desenvolvimento de Software

Criação dos Cenários de Teste de Aceitação

Criação dos Cenários de Teste de Sistema

Codificação

Requisitos

Especificação

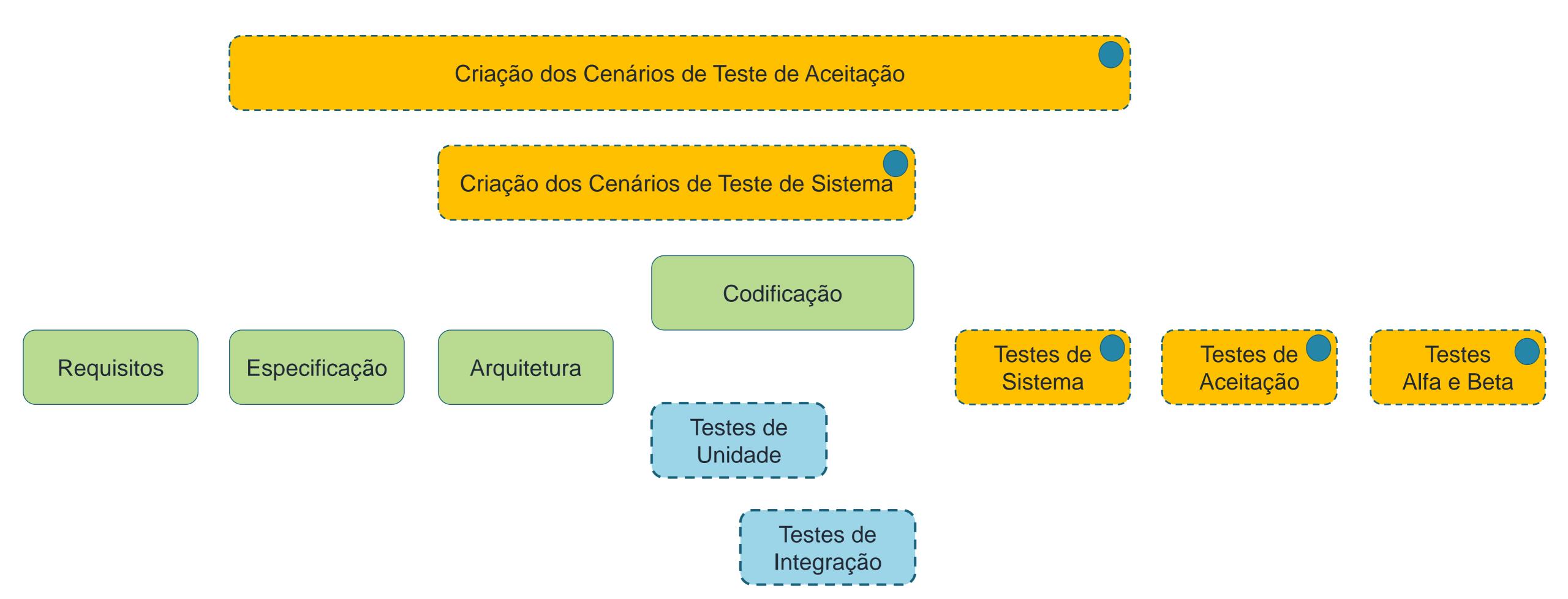
Arquitetura

Testes de Unidade

Testes de Integração Testes de Sistema

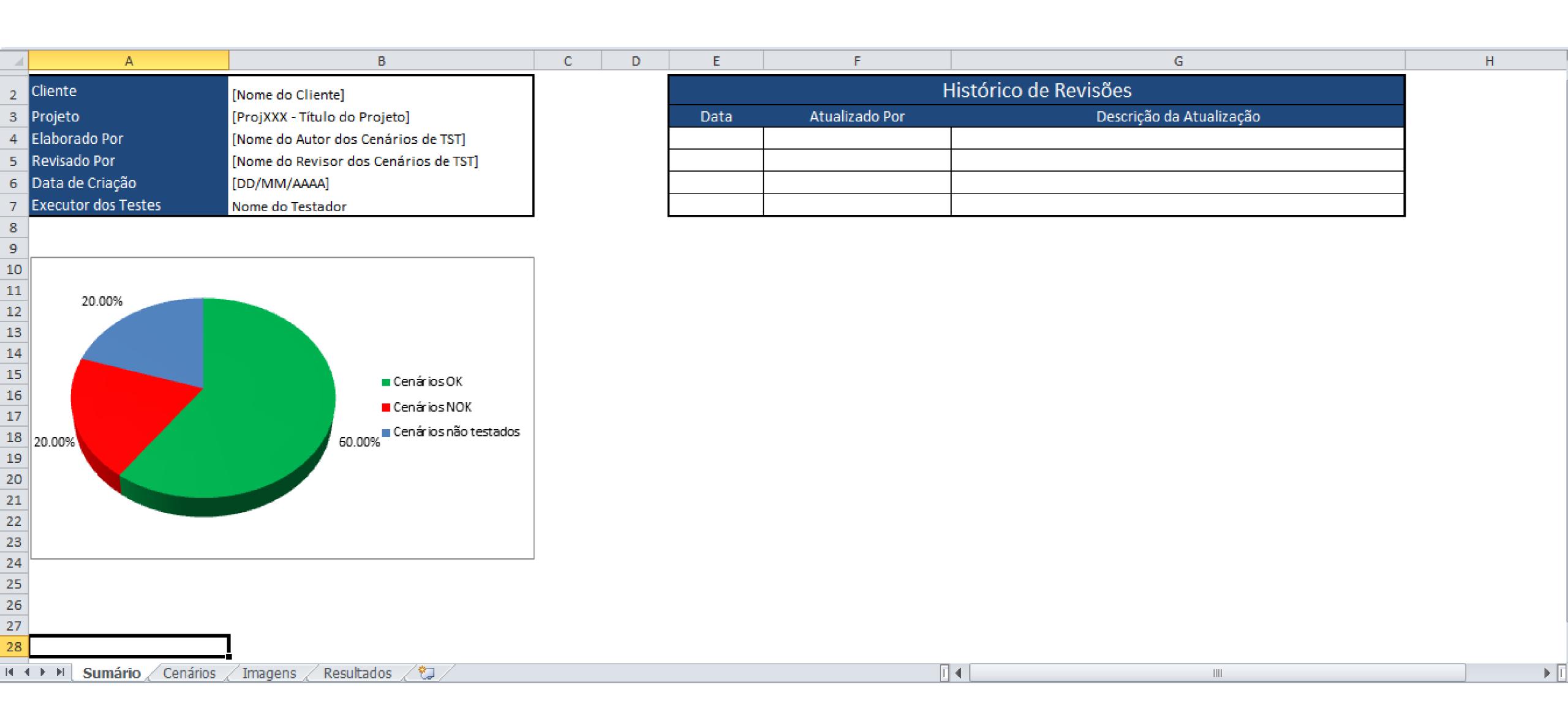
Testes de Aceitação

Testes Alfa e Beta



Criação do Plano de Testes

Ferramentas – Testes de Sistema



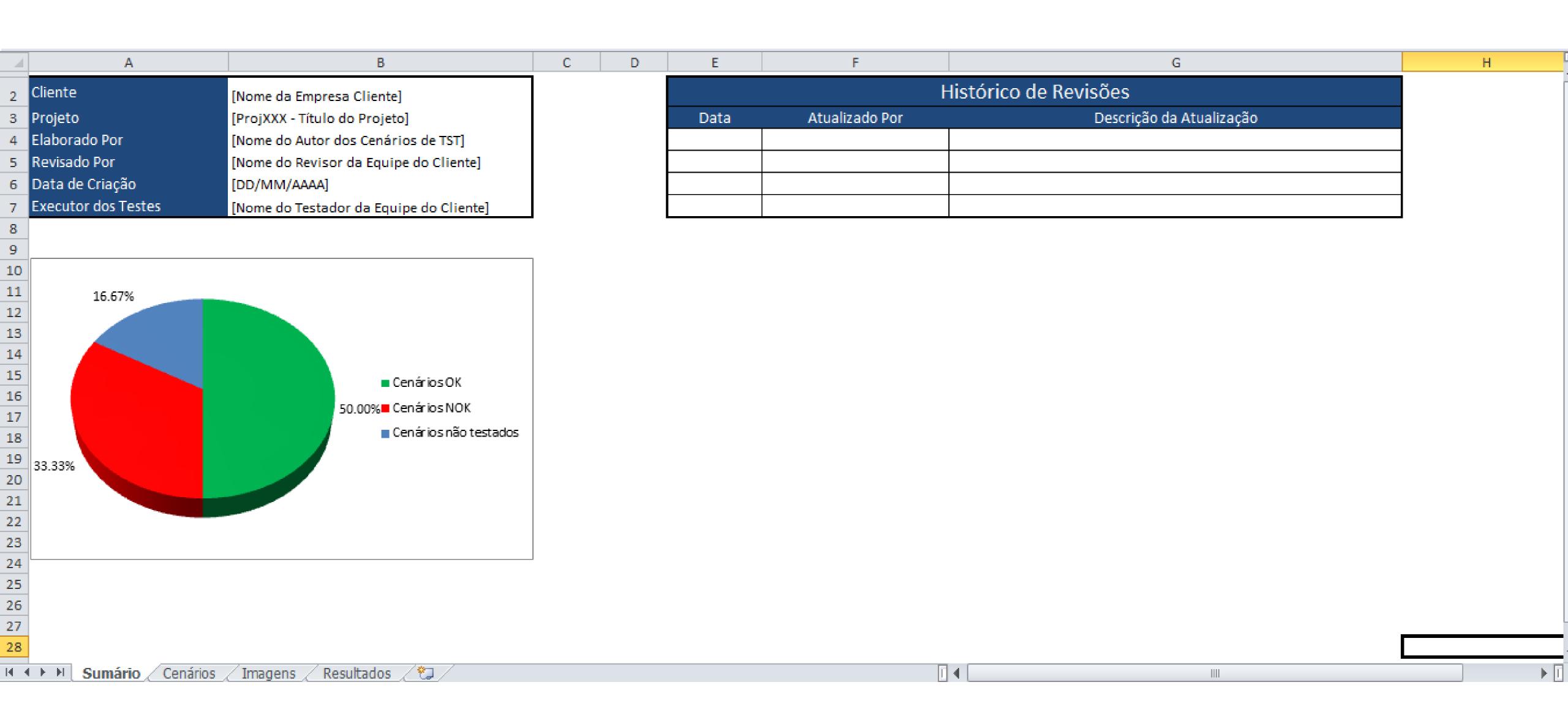
Ferramentas – Testes de Sistema

- 4	Α	В	С	D	E		G
1	ID.	Funcionalidade/ Componente	Tìtulo	Passo-a-passo para execução	Resultado esperado		Observação
2	1	Cadastro de Produto	Cadastro de produto sem nome	1- Efetuar login; 2- Na home do usuário, clicar no menu "Cadastros -> Cadastrar Novo produto"; 3- Preencher dados válidos para o novo produto, mas deixar o campo "Nome do Produto" em branco.	L- Usuário logar com sucesso e visualizar a sua home. 2- Deve ser aberta uma pop-up contendo um formulário para cadastro de um novo produto; 3- O sistema não deve permitir o cadastro do produto. Deve ser exibida uma nensagem de erro informando que este campo é obrigatório.		
3	2	Cadastro de Produto	1- Efetuar login; 2- Na home do usuário, clicar no menu "Cadastros -> Cadastro Novo produto"; 3- Preencher dados válidos para o novo produto, mas deixar o campo "Preço" em branco. 1- Efetuar login; 2- Deve ser aberta uma pop-up contendo um formulário para cadastro de um novo produto; 3- O sistema não deve permitir o cadastro do produto. Deve ser exibida uma mensagem de erro informando que este campo é obrigatório.		NOK		
4	3	Homepage	Validação de Menu	1- Acessar home do site: http://www.ilion.com.br/layouts/hf/ 2- Acessar o menu "Sobre a HF"; 3- Acessar o menu "Incorporação"; 4	2- Deve ser exibido um submenu com os seguintes itens: - missão - visão - valores	ОК	
5	4	Cálculo de Frete	Validar Frete quando temos vários produtos diferentes	1º prod: um celular com 350g 2º prod.: 2 capas p/ celular com 20g cada	Verificar que o frete deu R\$ 35		
4 4	5	Calculo de Preço		1- Efetuar Login 2- Adicionar ao carrinho 5 produtos com preços diferentes; 3- Clicar no ícone do carrinho de compras; 4- Inserir o CEP e clicar no botão	2- Verificar que, no canto superior direito da tela, ao lado do ícone do carrinho de compras, o contador de itens estará sendo atualizado até chegar em 5; 3- O usuário visualizará uma tela que exibirá os 5 itens adicionados ao carrinho e seus devidos preços. 4- O valor do frete deverá aparecer no campo "Valor do Frete" e também		

Ferramentas – Testes de Sistema

4	Α	В	С	D	Е	F	G	Н	T	J	K	L	М
1													
2		Resultados											
	Total de Cenários	5	100.00%										
	Cenários OK	3	60.00%										
	Cenários NOK	1	20.00%										
	Cenários não testados	1	20.00%										
7													
8													
9 10 11 12 13													
10													
11													
12													
13													
14													
16													
17													
18													
19													
20													
21	▶ ▶ Sumário / Cenários / Imag												
22													
23													
24												<u> </u>	
14 4	▶ № Sumário / Cenários / Imag	ens Resultados 💝			[] ∢				IIII) + [

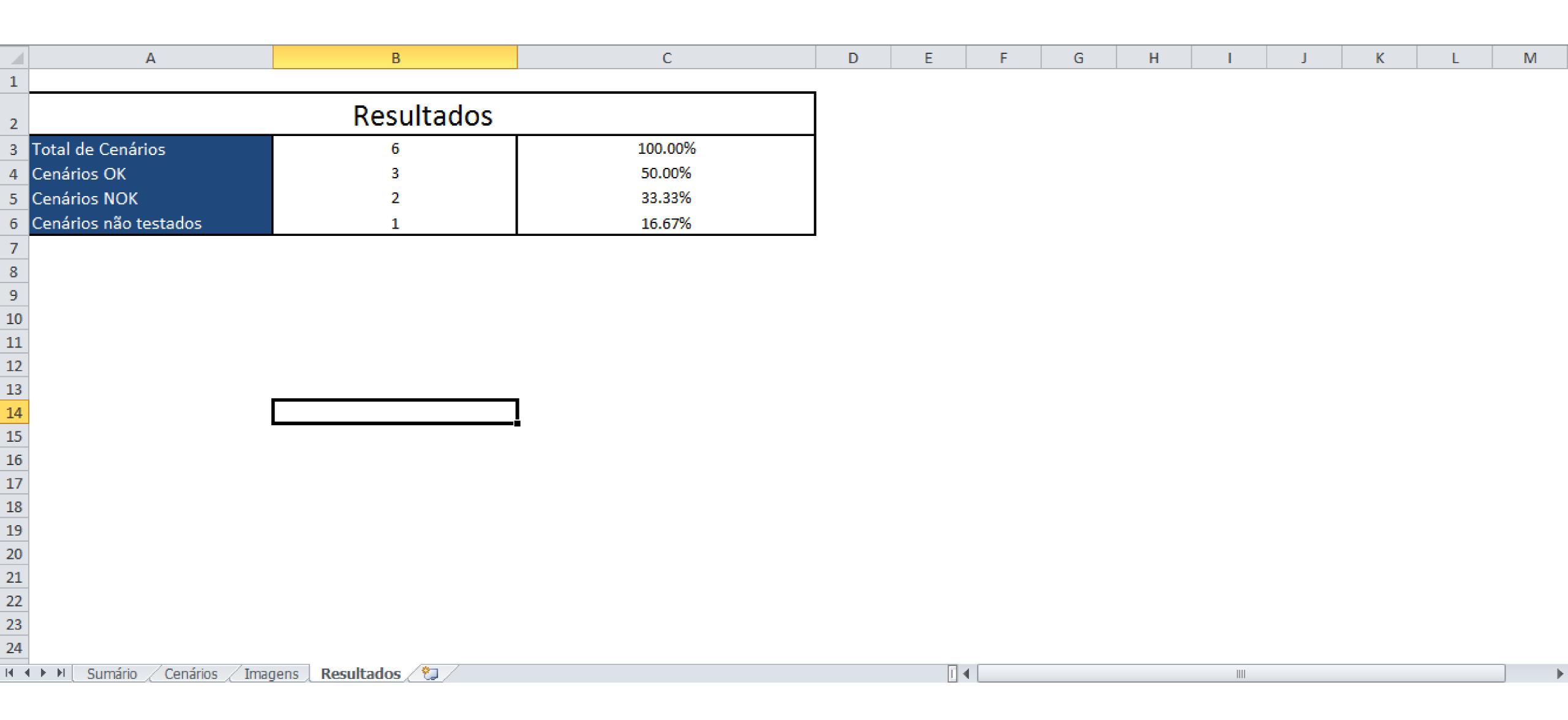
Ferramentas – Testes de Aceitação



Ferramentas – Testes de Aceitação

4	Α	В	С	D	E	F	G
1	ID.	Requisito	Operação a Ser Realizada	Resultado Esperado	Comentário do Cliente	Status	Testador (cliente)
2	1	Cadastro de Produto	Cada produto deverá ter um	O produto deve ser cadastrado no sistema	Funcionou e achei a interface legal, mas achei o sistema meio lento.	ОК	
3	2	Cadastro de Produto	ticarà inicialmente como não	Imediatamente	Legal, funcionou!	NOK	
4	3	Definição de imagem da empresa		Após entrar no sistema Admin e modificar a imagem principal da homepage, a antiga imagem deverá ser substituída de imediato.		ОК	
5	4	Cadastro de clientes				ОК	
6	5	Atualização de preços				NOK	
7	6	Atualização de formas de pagamento				-	
8	7 ► ►	Sumário Cenários Ima	ngens / Resultados / 知			-	

Ferramentas – Testes de Aceitação



Não técnicas:

- A pessoa que especifica os cenários de teste de sistema não deve estar participando da codificação do sistema
 - De preferência, que seja um testador especialista e não um programador
 - A especificação é a base para os cenários de teste
 - "O que não está especificado, não será validado"
- Faça uma boa seleção de cenários, considerando:
 - O tempo disponível para execução dos testes
 - A maturidade do(s) testador(es)
 - A relevância das funcionalidades a serem testadas
 - Lembre-se: os cenários são infinitos. Nunca haverá tempo para testar todos.
- Seja pessimista! Cuidado para não ignorar cenários aparentemente inocentes
- Registre todas as falhas encontradas e acompanhe até que seja corrigida

Técnicas:

- Não execute cenários roboticamente. Explore quando necessário.
- Achou uma falha? Detalhe ao máximo!
 - "A caixa de mensagens não segue o padrão"
 - "Após não preencher um dos campos obrigatórios no formulário de cadastro de usuário, a caixa de mensagens apresentada está em um formato diferente das demais caixas de mensagens de erro do sistema"
 - Browser: Chrome v10.2
 - Dispositivo: Tablet Ipad 7
 - (colocar printscreen)
- A falha foi corrigida? Reteste! (teste de regressão)
- Busque padrões ao reportar uma falha
 - Ex.: verifique se o erro ocorrido com um produto está associado apenas a tal produto ou se envolve todos os produtos de mesma categoria, cor, faixa de preço, etc.

• Técnicas:

- No lugar planilha que disponibilizei, você pode optar por ferramentas web para gestão de incidentes/falhas/melhorias identificas
 - Proj4me
 - Jira
 - Bugzilla
 - Trac
 - Etc.

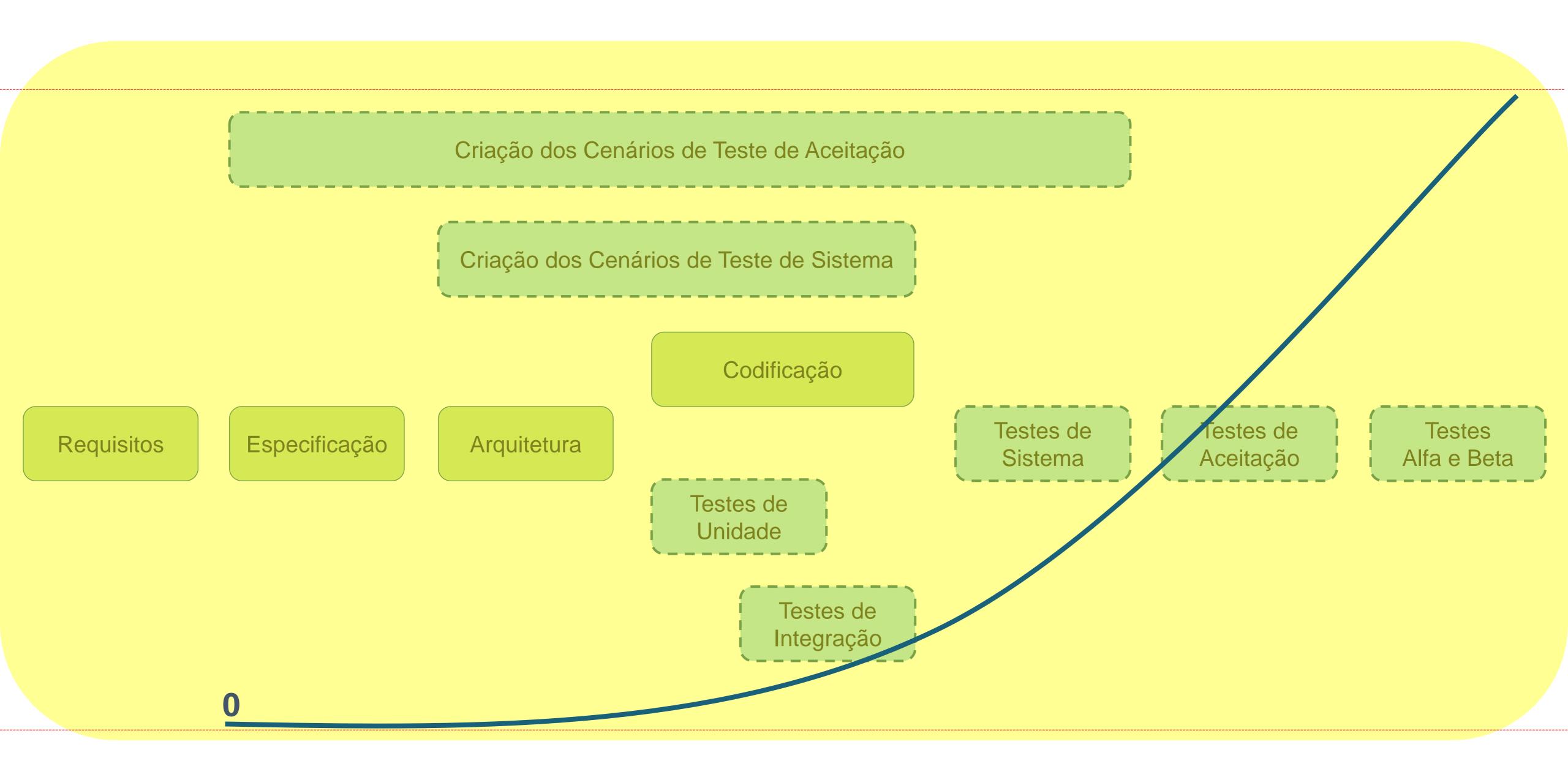
• Técnicas:

- Teste sempre os valores limites, grande parte dos erros estarão lá!
 - Formulários:
 - Campo em branco
 - Campo com valor inválido
 - Quantidade de caracteres
 - E-mail inválido
 - Mensagens de erro coerentes

Cálculos:

- Valores positivos
- Valores negativos
- ZERO
- Intervalos fechados e abertos
- Acima do limite máximo e abaixo do limite mínimo
- O mais importante: tente pensar como um usuário pensaria

O Custo da Falha



Mitos do Teste de Software

Mitos do Teste de Software

- Testando vamos encontrar 100% dos defeitos
- O teste de software é um custo extra para o projeto
- Qualquer pessoa é capaz de testar bem, é só seguir o plano de testes
- Programadores não precisam testar
- Os testes dos programadores são suficientes
- Só podemos testar quando o sistema estiver 100% pronto



- ISTQB / BSTQB: CTFL (Certified Tester Foundation Level)
 - www.istqb.org
- ALATS: CBTS (Certificação Brasileira de Teste de Software)
 - www.alats.org.br

A segunda oportunidade:

"Como transformar o aprendizado Deste curso em grandes oportunidades para sempre"

A segunda oportunidade

Empreendedor / Executivo / Professor / Vida / ?

Atencioso / Criterioso / Planejador / Detalhista / Oportunidades

Gerente de Projetos



Aplicação em nível administrativo (grandes projetos)

Analista de Sistemas



Referência e treinamentos internos

Graduação



Referência, ferramentas e artigos (em paralelo à programação)

A segunda oportunidade

Empreendedor / Executivo / Professor / Vida / ?



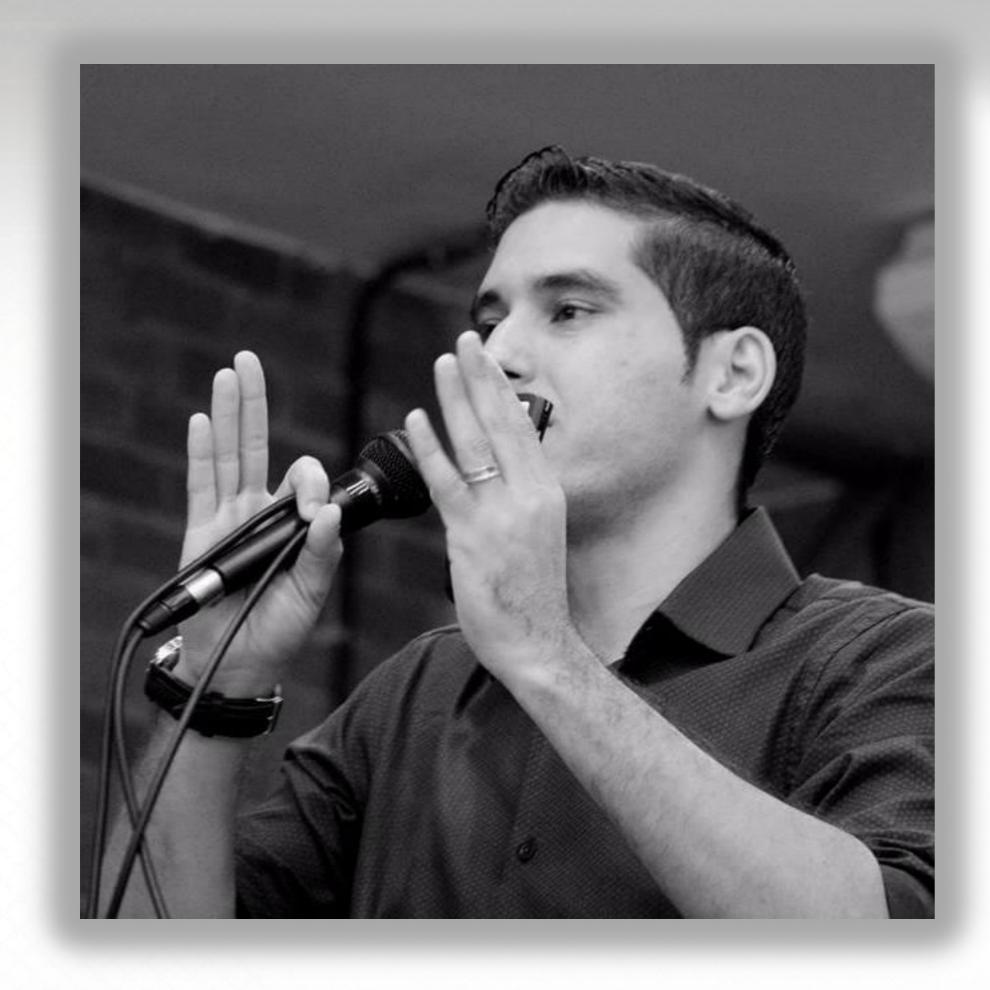
Lembre-se: você será reconhecido proporcionalmente ao tamanho do problema que você resolve e à sua raridade

Graduação



Falando em oportunidade...

Quer mais?



Teste de Software

Gustavo Farias

gustavo@vocegerente.com facebook.com/g.u.g.a.f.a.r.i.a.s.l.i.m.a

Principais Fontes

- 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.