



# Tipos de teste de software

---

Volnys Borges Bernal  
volnys@lsi.usp.br

Adilson Hira  
ayhira@lsi.usp.br

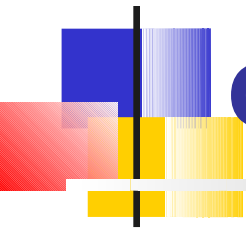
Laboratório de Sistemas Integráveis  
Departamento de Sistemas Eletrônicos  
Escola Politécnica da USP



# Sumário

---

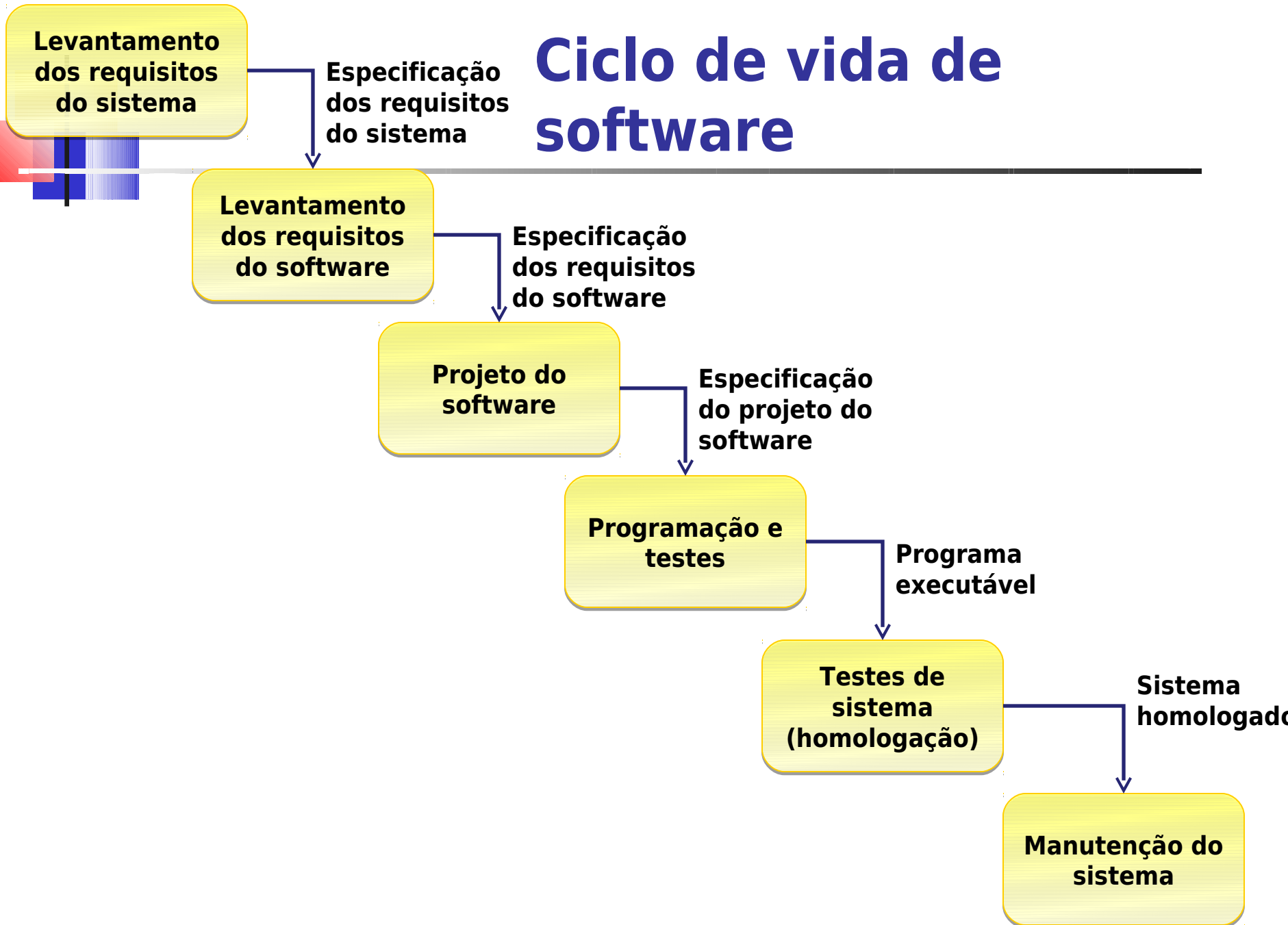
- Visão geral do ciclo de vida de software
- Teste em software
- Classificação dos testes de software



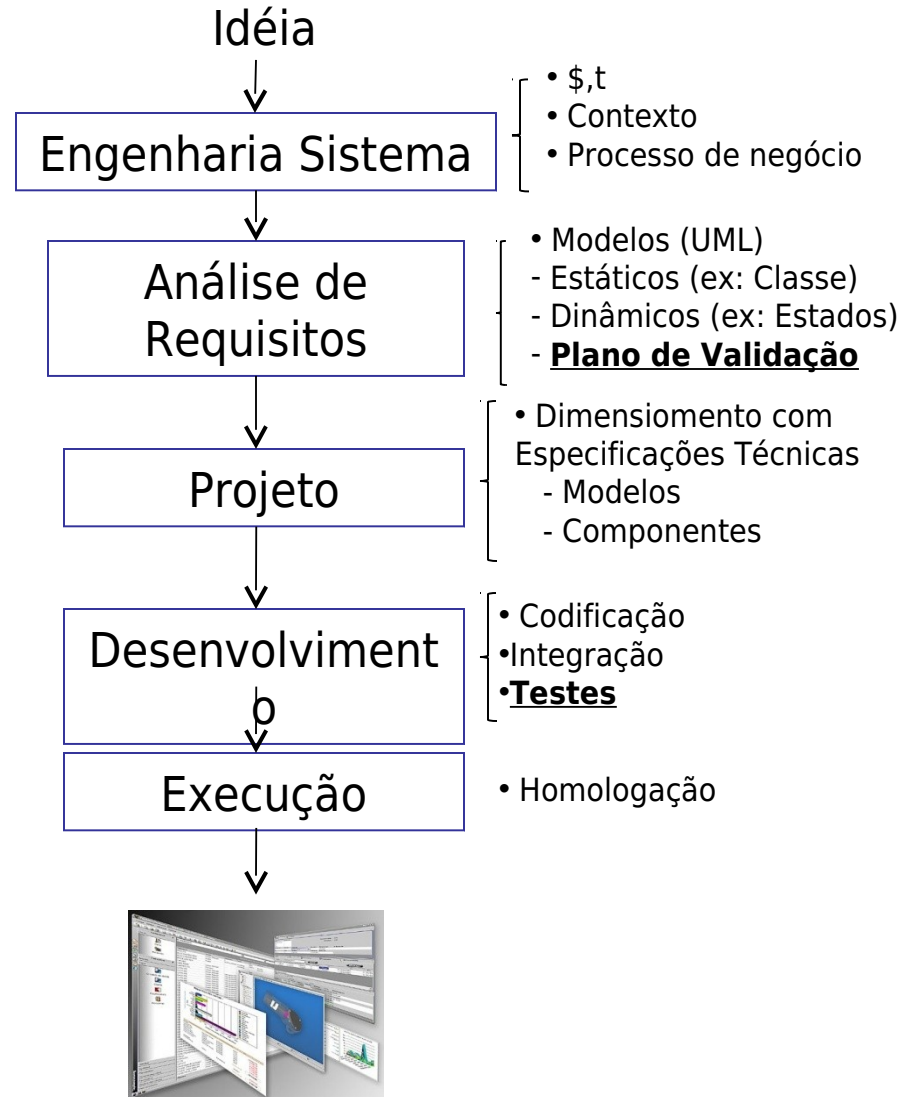
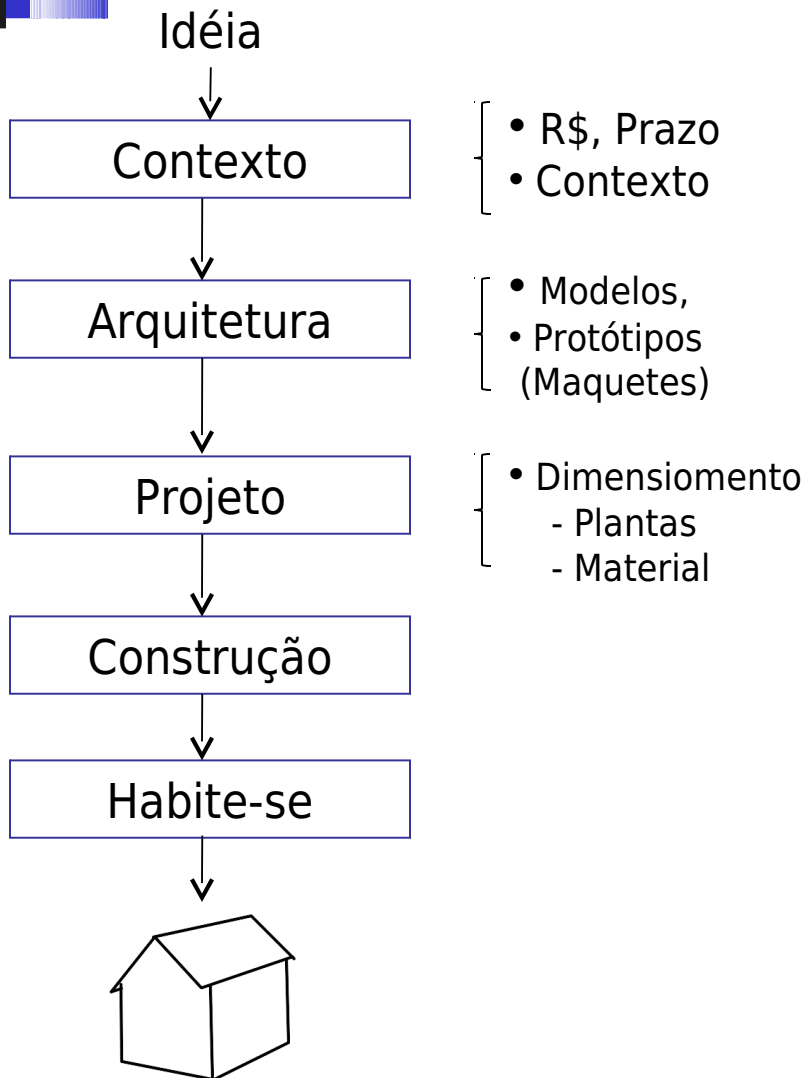
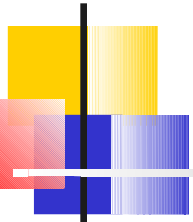
# Ciclo de vida de software

---

# Ciclo de vida de software



# Projeto de Civil X Projeto de Software





# Testes em software

---



# Testes em software

---

- Existem várias normas que tratam sobre teste em software e em sistemas:
  - IEEE 829 (Standard for Software and System Test Documentation)
  - ISO/IEC 29119 (Software Testing Standart)
  - ISTQB (International Software Testing Qualifications Board)



# Testes em software

---

- International Software Testing Qualifications Board
  - ISTQB
  - Fundada em novembro de 2002
  - Organização sem fins lucrativos voltada a capacitação de profissionais na área de teste
  - Define:
    - Conceitos utilizados na área de teste
    - Metodologias de testes para software
  - Ministra cursos
  - Realiza certificação de profissionais





# ISTQB

- <http://www.istqb.org/>

[FAQs](#) | [Contact us](#)

[Home](#) | [About us](#) | [ISTQB® in the World](#) | [Certification Path](#) | [Exams](#) | [News/Events](#) | [Get Involved](#) | [Downloads](#) | [References](#)

## Certifying Software Testers Worldwide

The ISTQB® has built the most renowned scheme for certifying software testers.

As of March 2012, ISTQB® has issued over 240.000 certifications in more than 70 countries world-wide, with a current growth rate of approximately 10.000 new certifications being issued per quarter.

The scheme relies on a Body of Knowledge (Syllabi and Glossary) and exam rules that are applied consistently all over the world. Material and exams are available in many languages.



## Events & Promotions



Choose an ISTQB® Board



News



- <http://www.bstqb.org.br/>

GLOSSÁRIO DE TERMOS | SYLLABUS FOUNDATION LEVEL | SYLLABUS ADVANCED LEVEL



BRAZILIAN SOFTWARE TESTING QUALIFICATIONS BOARD

HOME	A CERTIFICAÇÃO	SYLLABI	TREINAMENTO	PERGUNTAS FREQUENTES	GLOSSÁRIO	CONTATO
------	----------------	---------	-------------	----------------------	-----------	---------



243.451

Certificados  
no Mundo



2.122

Certificados  
pelo BSTQB



LOGIN DO USUÁRIO

Usuário: \*

Senha: \*

RECADO AOS CANDIDATOS

## Atenção candidatos.

**O exame terá início exatamente às 9hs de Brasília em todas as localidades. (Cuidado com o horário de verão).**

CALENDÁRIO DE EXAMES



Veja todas as datas referentes aos exames de certificação.



# Tipos de testes em software

---



# Tipos de teste de software

---

- Segundo o ISTQB, os testes podem ser classificados em função:
  - A) Do estágio do ciclo de vida do produto;
  - B) Do objetivo do teste;
  
- Obs:
  - Syllabus – base organizada de conhecimento



# Tipos de teste de software

---

A) Classificação dos testes em função do estágio do ciclo de vida do produto:

- A.1) Teste de componente;
- A.2) Teste de integração;
- A.3) Teste de sistema;
- A.4) Teste de aceite; e
- A.5) Teste de manutenção.



# Tipos de teste de software

---

B) Classificação dos testes em função do objetivo do teste:

- B.1) Teste funcional;
- B.2) Teste não funcionais;
- B.3) Teste estrutural; e
- B.4) Teste de regressão.



# **A) Testes em função do estágio do ciclo de vida**

---



# A) Testes em função do estágio do ciclo de vida

---

- Classificação dos testes em função do estágio do ciclo de vida do produto:
  - A.1) Teste de componente;
  - A.2) Teste de integração;
  - A.3) Teste de sistema;
  - A.4) Teste de aceite; e
  - A.5) Teste de manutenção.





# A.1) Teste de componente

---



## A.1) Teste de componente

---

- Procura defeitos e verifica o funcionamento do software (ex: módulos, programas, objetos, classes, etc.) que são testáveis separadamente.
- Pode ser feito isolado do resto do sistema, dependendo do contexto do ciclo de desenvolvimento e do sistema.
- Pode ser utilizado simulador.
- Pode incluir teste de funcionalidade e características específicas não-funcionais (ex: recursos [ex: falta de memória] e testes de robustez, além de teste estrutural.



## A.1) Teste de componente

---

- Tipicamente, o teste de componente ocorre com acesso ao código que está sendo testado e no ambiente de desenvolvimento, assim como um framework de teste de unidade ou ferramenta de depuração “debugging”.
- Envolve o programador do código.
- Defeitos são normalmente corrigidos assim que são encontrados sem registrar formalmente tais eventos.



## **A.2) Teste de integração**

---



## A.2) Teste de integração

---

- Caracterizado por testar as interfaces entre os componentes, interações de diferentes partes de um sistema (integração como o sistema operacional, hardware ou interfaces entre sistemas).
- Pode haver vários níveis de teste de integração, ex:
  - Teste de integração entre componentes;
  - Teste de integração entre sistemas.
- A cada estágio da integração, os testadores concentram somente na integração propriamente. Por exemplo, o módulo A está sendo integrado com o módulo B o foco é a interação entre os módulos, não suas funcionalidades. Tanto testes funcionais quanto estruturais podem ser utilizados.



## **A.3) Teste de sistema**

---



## A.3) Teste de sistema

---

- Trata o comportamento de todo do sistema ou produto definido pelo escopo de um projeto ou programa de desenvolvimento.
- Envolve os requisitos funcionais e não-funcionais
- No teste de sistema, o ambiente de teste deve ser o mais semelhante possível ao ambiente de produção, a fim de maximizar a identificação de falhas específicas de ambiente.
- Os testes de sistema podem ser baseados em especificação de riscos e/ou de requisitos, processos de negócios, casos de uso, dentre outras descrições de alto nível do comportamento, interações e recursos do sistema.



## A.3) Teste de sistema

---

- O teste de sistema deve tratar os requisitos funcionais e não-funcionais do sistema. Testadores devem também lidar com requisitos incompletos ou não documentados.
- Para o teste de sistema é ideal a utilização de uma equipe de teste independente.





## **A.4) Teste de aceite**

---



## A.4) Teste de aceite

---

- Freqüentemente é de responsabilidade do cliente ou do usuário do sistema; os interessados (*stakeholders*) também podem ser envolvidos.
- O objetivo do teste de aceite é estabelecer a confiança no sistema, de uma parte do sistema ou uma característica específica do sistema. Procurar defeitos não é o principal foco do teste de aceite.
- O teste de aceite pode ser realizado em diferentes níveis:
  - Nível de componente;
  - Nível do sistema.



## A.4) Teste de aceite

---

- Formas de teste de aceite
  - a) Teste de aceite de usuário
    - Objetivo: verificar se o sistema está apropriado para o uso por um determinado perfil de usuário.
  - a) Teste operacional de aceite
    - Objetivo: verificar a realização de atividades de operação pelo usuário administrador:
      - Teste de Backup/Restore.
      - Recuperação de desastre.
      - Gerenciamento de usuário.
      - Tarefas de manutenção.
      - Checagens periódicas de vulnerabilidades de



## A.4) Teste de aceite

---

- c) Teste de aceite de contrato e regulamento
  - Verifica se algum critério de aceite incluso em contrato na produção de software sob encomenda.
  - O critério de aceite deve ser definido quando o contrato é assinado. Teste de aceite de regulamento é quando se verifica a necessidade de adesão a algum regulamento de acordo com outras normas (ex: segurança, governamental, legislação).



## A.4) Teste de aceite

---

- d) Alfa e beta teste (ou teste no campo)
- Desenvolvedores de softwares comerciais ou pacotes, muitas vezes precisam obter um feedback de clientes em potencial existente no mercado antes que o software seja colocado à venda comercialmente.
  - O alfa teste ou teste de aceite de fábrica é feito no “site” da organização em que o produto foi desenvolvido.
  - O beta teste, teste de campo ou teste de aceite no site é feito pelas pessoas em suas próprias localidades. Ambos os testes são feitos pelos clientes em potencial e não pelos desenvolvedores do produto.



## **A.5) Teste de manutenção**

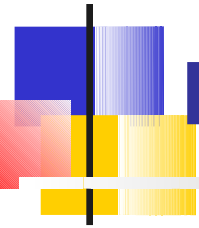
---



## A.5) Teste de manutenção

---

- Uma vez desenvolvido, um sistema pode ficar ativo por anos ou até mesmo décadas. Durante este tempo o sistema e seu ambiente podem ser modificados, corrigidos ou complementados.
- O teste de manutenção é realizado no mesmo sistema operacional, sendo realizado devido a modificações, migrações ou retirada de software ou sistema.
- Exemplos de modificações incluem melhorias planejadas (ex: baseadas em *releases*), mudanças corretivas e emergenciais, além de mudanças de ambiente, como atualização em sistema operacional ou banco de dados, e correções (*patches*) para expor e encontrar vulnerabilidades do sistema operacional



## **B) Testes em função do objetivo**

---





## B) Testes em função do objetivo

---

- B) Classificação dos testes em função do objetivo do teste:
  - B.1) Teste funcional;
  - B.2) Teste não funcionais;
  - B.3) Teste estrutural; e
  - B.4) Teste de regressão.



## **B.1) Teste funcional**

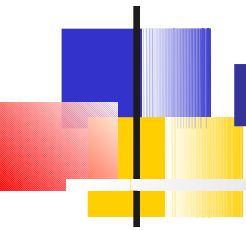
---



## B.1) Teste funcional

---

- São baseados em funções descritas em documentos de especificação ou compreendidas pelos testadores, e devem ser realizados em todos os níveis de teste (ex: teste de componente).
- As técnicas baseadas em especificação podem ser utilizadas para derivar as condições de teste e casos de testes a partir da funcionalidade do software ou sistema. O teste funcional considera o comportamento externo do software (teste de caixa-preta).



## **B.2) Teste não funcional**

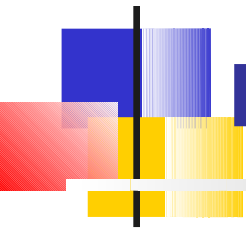
---



## B.2) Teste não funcional

---

- Inclui, mas não se limitam a:
  - teste de desempenho;
  - teste de carga;
  - teste de estresse;
  - teste de usabilidade;
  - teste de interoperabilidade;
  - teste de manutenibilidade;
  - teste de confiabilidade,
  - teste de segurança, e
  - teste de portabilidade.
- Testes não-funcionais podem ser realizados em todos os níveis de teste.



## **B.3) Teste estrutural**

---



## B.3) Teste estrutural

---

- O teste estrutural, também chamado de teste de caixa-branca, é uma técnica de teste que serve para avaliar o comportamento interno do componente de software.
- O teste é realizado diretamente sobre o código fonte dos componentes de softwares para que possam ser avaliados os aspectos como os teste de fluxo de dados, teste de condição, teste de ciclos e os teste de caminhos lógicos.
- Os testes estruturais podem ser realizados em todos os níveis de testes. Recomenda-se utilizar as técnicas estruturais após as técnicas baseadas em especificação.

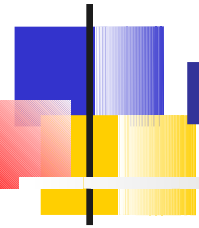


## B.3) Teste estrutural

---

- Exemplos de testes estruturais
  - Uso da ferramenta livre **JUnit** para desenvolvimento de classes de teste para testar classes ou métodos desenvolvidos em **Java**.
  - Técnica testes manuais ou testes efetuados com apoio de ferramentas para verificação de aderência a boas práticas de codificação





## **B.4) Testes de regressão**

---



## B.4) Testes de regressão

---

- O teste de regressão consiste na aplicação de testes à versão mais recente do software, para verificar que não existem novos defeitos em componentes já testados. Se, ao juntar um novo componente ou suas alterações a outros **componentes** do sistema surgirem novos defeitos em componentes inalterados, então considera-se que o sistema regrediu.
- Os testes de regressão, também chamados testes decorrentes de mudanças são os testes realizados novamente em um programa que já foi testado, após sua modificação, para descobrir a existência de algum defeito introduzido ou não coberto originalmente pela mudança.