



Executar Teste e Implantação de Aplicativos Computacionais

SENAC PE

24 de Setembro de 2024

Conceitos Básicos:

- Fases da Atividade de Teste
- Técnicas e Critérios de Teste



CASO 1

AIRBUS MODELO A320-211

- Em 1993, um avião (Airbus 320-211) operado pela Lufthansa saiu de Frankfurt com destino ao aeroporto de Arsaw.
- Condição do pouso: tempo ruim, chuva, vento forte com variação de velocidade, pista molhada
- Na aterrissagem, o piloto acionou o sistema de controle dos freios, mas nada aconteceu.
- Demorou 9 segundos, depois que o avião tocou o solo, para que o sistema de trava funcionasse.
- Não existia pista suficiente para que o avião parasse com segurança.
- Resultado: 2 mortos, 56 feridos.
- Uma das causas do acidente: **a especificação do software de controle de freio da aeronave não considerava aquelas condições de pouso encontradas.** O software concluiu que o avião não havia tocado o chão.
- Foco no software:
 - Se comportou da forma como foi especificado.
 - Não havia erros no software.
 - Exemplo de software confiável, mas inseguro.



Fonte: https://pt.wikipedia.org/wiki/Voo_Lufthansa_2904

CASO 2

ARIANE 5

- Em 4 de junho de 1996, na Guiana Francesa, falha no lançamento do foguete Ariane 5 (lançador descartável usado para colocar satélites artificiais em órbitas geoestacionárias).
- Após 40s, o lançador desviou o caminho de voo, quebrou e explodiu.
- Iniciou-se uma investigação para identificar se os **testes de qualificação** e os **testes de aceitação** eram apropriados em relação ao problema encontrado, e recomendar medidas corretivas para remover as causas da anomalia e outros possíveis lacunas do sistemas relacionados.
- O sistema de orientação parou de funcionar.
- A exceção interna do software de orientação foi causada durante a execução de uma conversão de dados de ponto flutuante de 64 bits para o valor inteiro assinado de 16 bits. O número de ponto flutuante que foi convertido teve um valor maior do que o que poderia ser representado por um inteiro de 16 bits. Isso resultou em um Erro de Operando.



Fonte: <https://www.geogebra.org/m/PgTqKVGu>

Ou seja...

Conclusão das análises dos casos

- A atividade de teste é complexa.
- Diversos são os fatores que podem ocasionar erros.
 - Caso Airbus (melhorar a política de segurança e previsão de erros, simulações, etc)
 - Caro Ariane 5 (problema no algoritmo que continha a função que fazia a conversão de valores).
- Ou seja, tipos distintos de engano e níveis diferentes de abstração.
- **Por isso, a atividade de teste é dividida em fases com objetivos distintos.**
- **Cada fase aborda diferentes tipos de erros e aspectos do software.**

As fases da Atividade de Teste

Fase	Foco	Erros que podem ser relatados	Observação
Teste de Unidade	Menores unidades de um software (funções, procedimentos, métodos, classes).	Erros simples de programação, estruturas de dados incorretas, algoritmos mal implementados ou incorretos.	Pode ser feito durante a implementação e pelo próprio desenvolvedor.
Teste de Integração	Construção da estrutura do software.	Interfaces entre as unidades.	Feito após testar as unidades. Executado pela equipe de desenvolvimento, que conhece a estrutura interna do software
Teste de Sistema	O Sistema .	Erros de funções e características de desempenho que não estejam de acordo com a especificação.	Feito após o sistema estar completo. Pode ser feito por uma equipe independente. Pode-se considerar requisitos funcionais e não funcionais (segurança, performance, etc).

Teste de **Regressão**

- Manutenções podem inserir novos defeitos.
- Por isso, este teste é feito **após as manutenções** do software, para garantir que as **modificações** estão corretas e os requisitos anteriormente testados **continuam válidos**.

Resumo



Pergunta

Questão sobre a Atividade de Teste

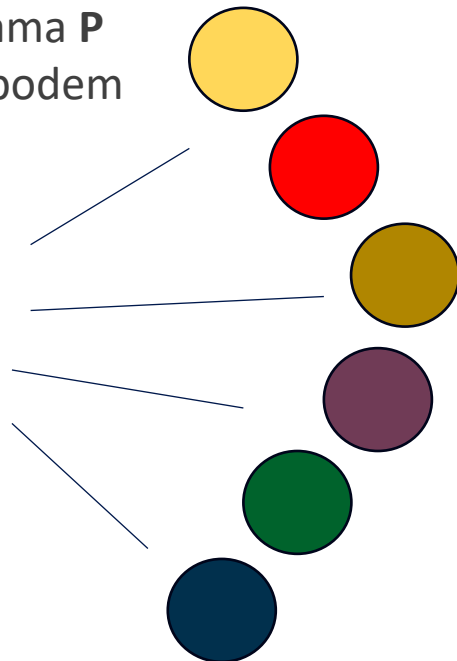
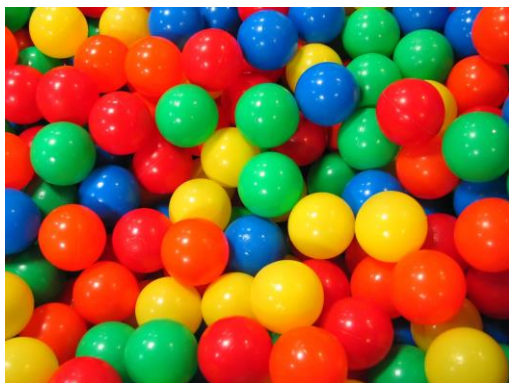
- A sua equipe, composta por você e mais dois desenvolvedores Python, está trabalhando na construção de um sistema para gerenciar publicações, projetos, notícias e eventos relacionados a grupos de pesquisa de uma universidade. Ainda não foram realizados testes no sistema, de forma alguma. O contato com o cliente é inviável neste momento, pois ele está viajando. Adicionalmente, nenhuma modificação foi solicitada por esse cliente até o momento.
- Considerando este cenário e a fase em que vocês se encontram, qual é a melhor opção a ser adotada?
 - A) Realizar Teste de Unidade, a fim de encontrar erros simples de programação e implementações que ainda estão ruins. Isso será feito ao longo da construção do sistema, por você e seus colegas desenvolvedores.
 - B) Esperar o cliente voltar da viagem e pedir que ele faça testes nas partes já implementadas. São poucas, mas quanto antes iniciar os testes, melhor. Vamos começar iniciando com a visão do cliente.
 - C) Realizar Teste de Integração, uma vez que vocês acreditam na experiência da equipe em desenvolver software e certamente nenhum erro menor foi inserido durante o desenvolvimento.

Conceitos Básicos

- Técnicas e critérios de teste

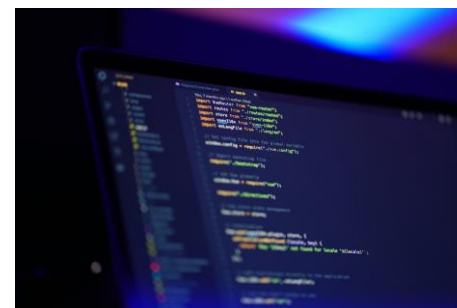
Cenário Típico da Atividade de Teste

Domínio de entrada de um programa **P**
(conjunto de todos os valores que podem
ser usados para executar P)



Dado de teste (elemento do domínio de entrada de P)

Caso de Teste
(caso de teste, resultado esperado)
Conjunto de casos
de teste



Programa P

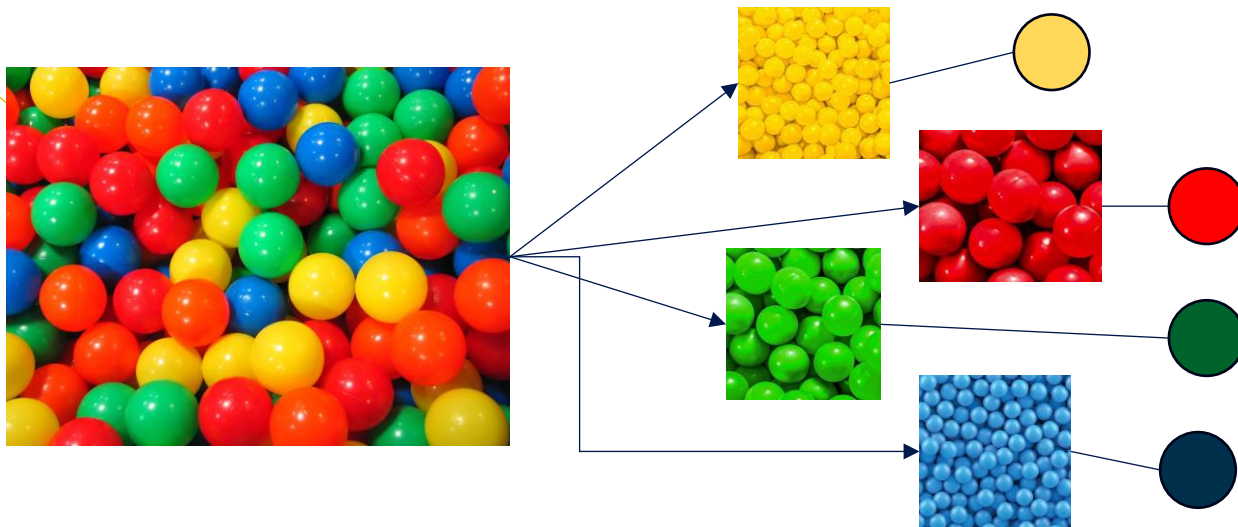
Saída
Obtida

São
iguais?

Saída
Esperada

Questões para pensar!

- O ideal seria testar o programa com todos os elementos do domínio de entrada, para garantir que não há defeitos (**teste exaustivo**).
- **Infactível** para grandes ou infinitos domínios de entrada.
- O que fazer? Selecionar **subdomínios**, mas garantindo a alta probabilidade em revelar defeitos no software, caso existam.



Como identificar os subdomínios?

- São definidos “**regras**”, chamadas de **técnicas** e seus critérios, para identificar quando dados de teste devem estar no mesmo domínio ou em subdomínios diferentes.

No exemplo anterior, a regra poderia ser a **cor**?

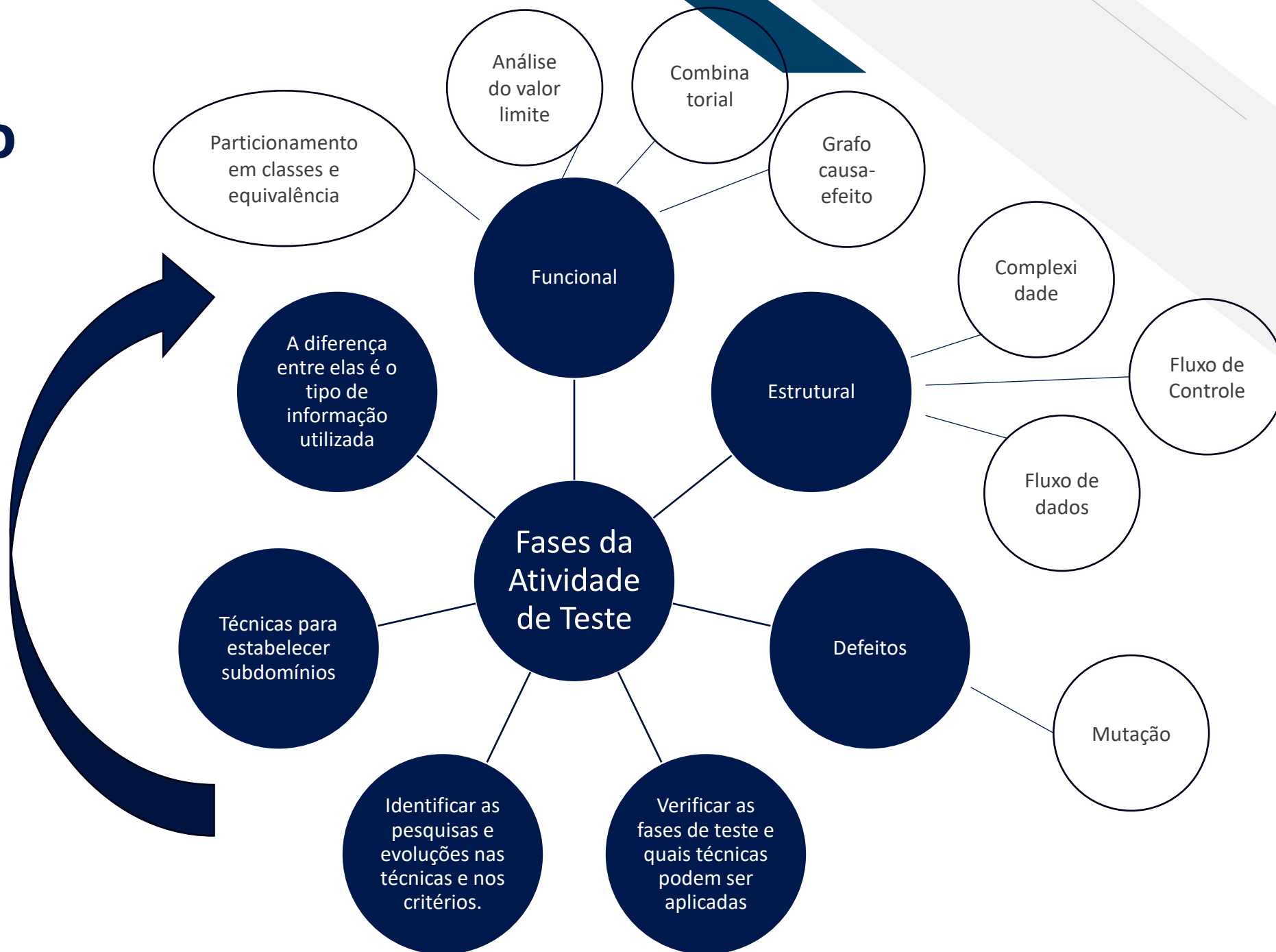
Mas como fazer isso com um **software**?

Técnicas de Teste

Técnicas	Tipos de informação Utilizada para derivar os Requisitos de teste	Passos principais	Exemplos de critérios
Teste Funcional	Especificação do software	a) Identificar as funções que o software deve realizar. b) Criar casos de teste capazes de verificar se tais funções estão sendo executadas corretamente	<ul style="list-style-type: none"> • Particionamento em Classes e Equivalência • Análise de valor limite • Combinatorial • Grafo de causa-efeito • Error-guessing • Teste funcional sistemático
Teste Estrutural	Conhecimento da estrutura interna do programa (implementação, código fonte)	A maioria dos critérios desta técnica utiliza uma representação de programa conhecida como grafo de programa.	<ul style="list-style-type: none"> • Critérios baseados em complexidade (McCabe) • Critérios baseados em fluxo de controle (todos nós, todos arcos, todos caminhos) • Critérios baseados em fluxo de dados (Rapps & Weyuker; Potenciais usos)
Técnicas baseada em defeitos	Defeitos típicos do processo de implementação de software.	Criar implementações alternativas e forçar o testador a projetar casos de teste que revelem os defeitos nelas induzidos.	Teste de Manutenção <ul style="list-style-type: none"> • Análise de Mutantes • Mutação de Interface

- **Funcional:** todas as fases de teste (unidade, integração, sistema).
- **Estrutural:** fases de teste de unidade e teste de integração.
- **Mutação:** fases de teste de unidade e teste de integração.

Resumo



Pergunta

Questão sobre as técnicas

- O uso de uma técnica não exclui o uso de outra. Ou seja, você pode usar várias técnicas para gerar os casos de teste.
- A escolha de qual técnica utilizar dependerá de uma série de fatores, por exemplo o tipo do sistema, clientes, requisitos contratuais, nível e tipo de riscos, objetivos do teste, documentação disponível, conhecimento dos testadores, tempo, dinheiro, ciclo de desenvolvimento adotado (cascata, iterativo/incremental, etc.), modelo de caso de uso e uma experiência prévia dos tipos de defeitos encontrados, dentre outras questões.
- Você e sua equipe receberam a tarefa de implantar o ScriptLattes, que é um script GNU-GPL desenvolvido para a extração e compilação automática de: produções bibliográficas, produções técnicas, produções artísticas, orientações, projetos de pesquisa, prêmios e títulos, grafo de colaborações e mapa de geolocalização de um conjunto de pesquisadores cadastrados na plataforma Lattes.
- Vocês possuem informações sobre a especificação do ScriptLattes (manual, funcionalidades, atores, etc.), mas não possuem documentação necessária sobre o código em si e não possuem conhecimento da estrutura interna, a qual foi criada por terceiros.
- Nesse cenário, qual seria a opção de Técnica de Teste mais apropriada para derivar os casos de teste?
- Teste de Mutação
- Teste Estrutural
- Teste Funcional



Senac Pernambuco Educação Profissional Recife

Thiago Dias Nogueira

Instrutor Técnico

(81) 9 9627-0419

thiago.nogueira@pe.senac.br