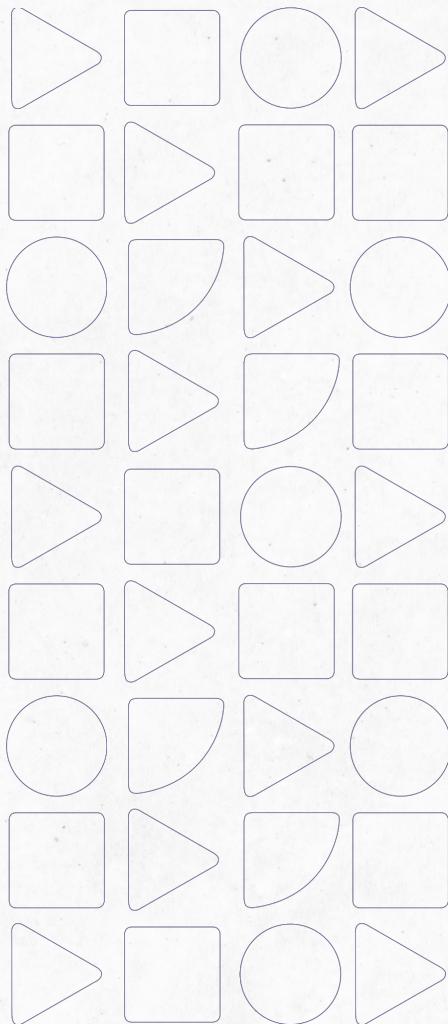


Testes funcionais

Disciplina: Qualidade de *Software*



Conteúdos:

Testes funcionais.

Habilidade(s):

Compreender como funcionam os testes funcionais em problemas aplicados.

Bloco 1

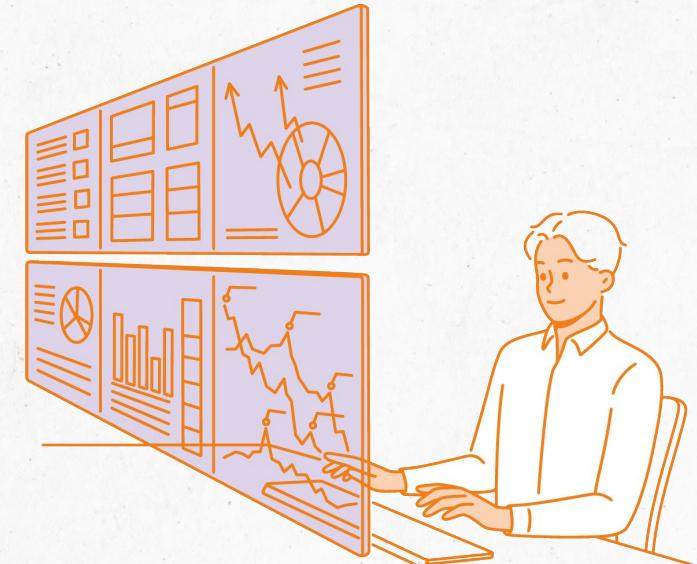
Compreendendo os testes funcionais.

Refletindo sobre a profissão

Se você fosse um avaliador de *software*, quais tipos de programas gostaria de testar? Por quê?



Os testes são uma parte essencial do processo de desenvolvimento de *software*. Eles ajudam a identificar defeitos, falhas e problemas antes do *software* ser lançado para os usuários finais.



Como os testes são feitos?

Você escolhe valores de entrada para testar o programa e cada um deles é um dado de teste.



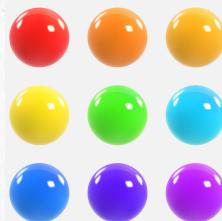
Domínio de entrada



Dados de teste

Como os testes são feitos?

- Cada dado de teste se torna, então, um caso de teste com a entrada e o resultado esperado;
- Você reúne todos os casos de teste em um conjunto de casos de teste;
- O conjunto de casos de teste é usado para testar o programa e verificar se ele funciona como se espera.



Dados de teste

Casos de teste

Conjunto de casos de
teste



Programa

Um pequeno impasse

Em domínios grandes ou infinitos, como testar todos os dados da piscina de bolinhas?



A resposta é:

Com as técnicas de teste! Na aula de hoje, conhceremos um pouco mais sobre os **testes funcionais**.



Testes funcionais

Concentram-se em verificar se o *software* realiza as suas funções de acordo com as especificações.

Isso envolve a criação de casos de teste que abordam diferentes cenários de entrada e de saída.

Suponha que você está testando uma calculadora que deve somar dois números. A seguir, veja alguns casos de teste funcionais:

- caso de teste: entrada (3, 5) - a saída esperada é 8 ($3 + 5$);
- caso de teste: entrada (0, 0) - a saída esperada é 0 ($0 + 0$);
- caso de teste: entrada (-2, 2) - a saída esperada é 0 ($-2 + 2$).



Principais características dos testes funcionais

É preciso identificar as funções e, a partir delas, criar casos de teste capazes de verificar se elas estão corretas.

Dados utilizados

Especificações do *software*, como recursos, funcionalidades específicas e ações.

Em que fase trabalha

Unidade, progressão e sistema.

Principais aspectos

Particionamento das classes de equivalência, análise do valor limite e combinatória.

Comparação com outros tipos de teste

Além dos testes funcionais, outros tipos de testes realizados, como os **estruturais** e os de **defeitos**. Confira a tabela a seguir.

Aspectos	Funcional	Estrutural	De defeitos
Dados utilizados	Especificação	Implementação	Defeitos típicos
Processo de teste	Identificar funções	Representação do programa	Implementação alternativa
Critérios	Particionamento das classes de equivalência, análise do valor limite, combinatória	Complexidade, fluxo de controle, fluxo de dados	Teste de mutação
Fase de teste	Todas	Unidade e integração	Teste de caixa preta

Bloco 2

Compreendendo os testes funcionais.

Quem sou eu?

Baseando-se nos testes apresentados no bloco anterior, identifique, de acordo com as descrições a seguir, a qual teste cada personagem abaixo se refere.



?

Seu conhecimento profundo
permite que ele identifique falhas e
erros.



?

Está sempre focada em garantir que
o *software* atenda os requisitos
definidos.



?

É especialista em encontrar agulhas
em palheiros de código.

Quem sou eu?

Baseando-se nos testes apresentados no bloco anterior, identifique, de acordo com as descrições a seguir, a qual teste cada personagem abaixo se refere.



Estrutural

Seu conhecimento profundo permite que ele identifique falhas e erros.



Funcional

Está sempre focada em garantir que o *software* atenda os requisitos definidos.



De defeitos

É especialista em encontrar agulhas em palheiros de código.

Vamos imaginar o seguinte cenário

Ao construir a casa dos seus sonhos, você precisa contratar especialistas para se assegurar de que a construção atenda às suas necessidades.

Certamente, você faria inspeções regulares para verificar os aspectos externos, conferindo se a cor da tinta escolhida na loja ficou bonita nas paredes da casa, se há rachaduras ou se existem discrepâncias em relação às suas expectativas.



O teste funcional é como construir uma casa!

Ele é como os especialistas que realizam as inspeções externas na sua casa, garantindo que tudo está correto e de acordo com as suas **especificações**.

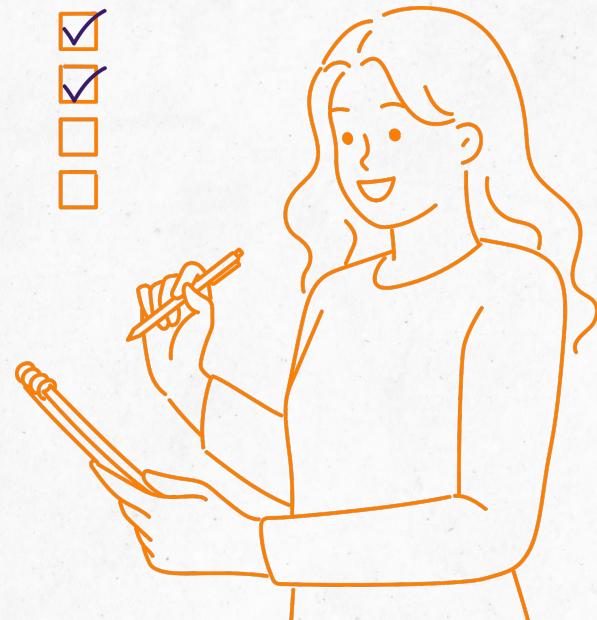


O que é uma especificação?

É um documento detalhado que descreve as características, as funcionalidades, os requisitos e os comportamentos esperados de um *software*.

Tente responder:

- o usuário consegue fazer tal coisa no sistema?
- tal requisito é adequado e funciona da forma que deveria, isto é, como foi identificado antes?
- como resultado, o que temos?
- são reveladas inconformidades com os objetivos especificados anteriormente. Quais tipos de inconformidades?



Permite equipes interdependentes

Isso se relaciona com a metodologia de **desenvolvimento ágil**.

Nesse contexto, equipes independentes são responsáveis por desenvolver e entregar as funcionalidades completas do *software*, ou seja, elas não se concentram em camadas ou em componentes específicos.



Bloco 3

O que é particionamento em classes de equivalência?



Imagine que você está testando um programa que possui um domínio muito alto.

Você deve testar todos os dados, garantindo a sua eficácia, ou deve dividir em partes menores e representativas?



Imagine que você está testando um programa que possui um domínio muito alto.

**Isso mesmo, é preciso dividir em partes menores e representativas!
Quando se trata de testes exaustivos, essa é a melhor saída.**

Particionamento de classes de equivalência

É utilizado para reduzir o espaço de entrada de um sistema a ser testado, objetivando criar casos de teste significativos e representativos.

Em vez de testar cada possível valor de entrada, divide-se o conjunto de todas as entradas possíveis em grupos ou classes tratados de maneira similar pelo sistema.

Isso ajuda a simplificar o processo de teste, tornando-o mais focado e gerenciável.



Preste atenção no princípio: se um caso de teste funciona para um elemento de uma classe, ele também deverá funcionar para todos os outros elementos dessa mesma classe.



Como isso funciona?

Objetivo

Dividir o conjunto de todas as entradas possíveis em grupos ou classes que são tratados de maneira similar pelo sistema.

Identificação de classes de equivalência: a partir da especificação do *software*, você identifica as variáveis de entrada e suas condições.

Definição de casos de teste: para cada classe de equivalência identificada, você define casos de teste, selecionando um representante da classe.

Seleção de valores válidos e inválidos: para cada classe de equivalência, você define um caso válido que atenda às condições da classe e aos casos inválidos que não atendam a essas condições.

Execução dos casos de teste: os casos de teste gerados a partir das classes de equivalência, então, são executados no *software* a ser testado.

Vamos observar o exemplo



Considere um *software* que calcula o Índice de Massa Corporal (IMC) de uma pessoa com base em sua altura e peso.

Identificação de classe de equivalência: a altura deve ser maior do que zero e deve ter até três metros. O peso, por sua vez, deve ser maior do que zero e de até 300 quilogramas.

Definição de casos de teste

1. Altura: 1.75 (altura válida), peso: 70 (peso válido) e resultado esperado: cálculo do IMC bem-sucedido;
2. Altura: -0.5 (altura inválida), peso: 65 (peso válido) e resultado esperado: erro de entrada inválida;
3. Altura: 2.2 (altura válida), peso: 350 (peso inválido) e resultado esperado: erro de entrada inválida.

Para cada classe de equivalência, selecionamos casos válidos e inválidos. A partir disso, quais testes passarão durante a execução?

Bloco 4

Conhecendo a análise do valor limite.

Vamos recapitular o que vimos?

Você deve ter conhecido o **particionamento de classes de equivalência**. Considerando os casos abaixo, diga como você realizaria a **identificação de classes de equivalência**, a **definição de casos de teste** e a **seleção de valores válidos e inválidos**.

a

Imagine que você está testando um sistema de autenticação que verifica a validade das senhas dos usuários.

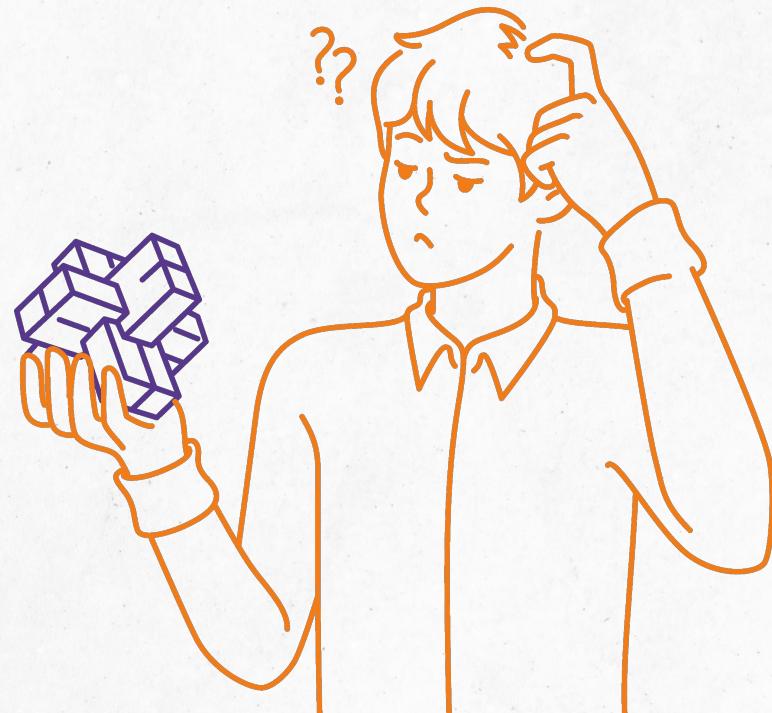
b

Vamos considerar um sistema de *e-commerce* que permite o cadastro de produtos por vendedores.

c

Considere um sistema de reservas de passagens aéreas.

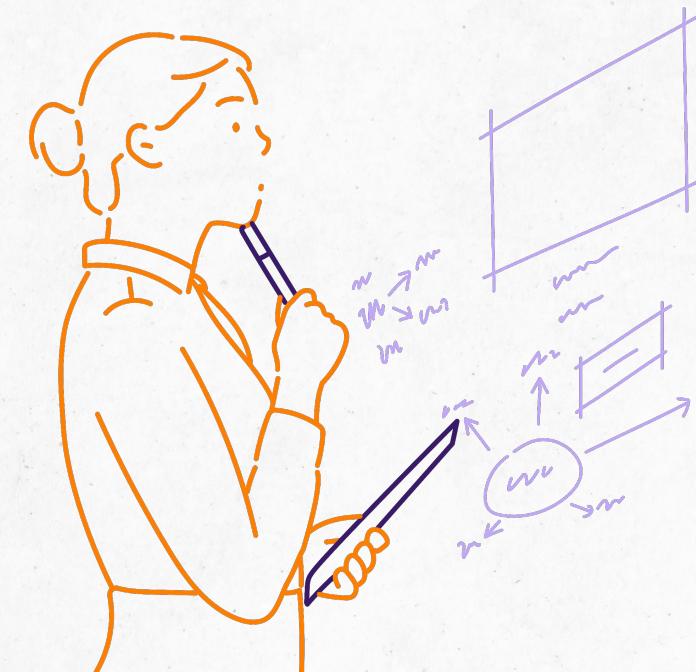
Depois da etapa de particionamento de classes de equivalência, a análise do valor limite é responsável por testar os cenários que envolvem os limites das classes de equivalência identificadas. Ela vai um passo adiante, focando nos pontos extremos desses conjuntos.



Análise do valor limite

Envolve testar os valores que estão exatamente nos limites das classes de equivalência, o que inclui os limites inferior e superior de cada classe.

Esses testes buscam verificar se o *software* se comporta corretamente quando os valores estão no limite do aceitável.



Vamos observar o exemplo

Suponha que você está testando um sistema de cadastro de usuários para um *site*. Nele, os usuários podem definir a idade.

A faixa aceitável para registro é de 18 até 65 anos. Desse modo, você deseja aplicar a análise de valor limite, garantindo que o sistema lide corretamente com os limites etários.

Limite inferior

Entrada: 18 anos.

Esperado: o sistema deve aceitar a entrada e permitir que o usuário seja registrado, uma vez que 18 anos é o limite inferior da faixa aceitável.

Limite superior

Entrada: 65 anos.

Esperado: o sistema deve aceitar a entrada e permitir o registro, pois 65 anos é o limite superior da faixa aceitável.

Fora do limite

Entrada: 17 anos.

Esperado: o sistema deve rejeitar a entrada e exibir uma mensagem de erro, informando que a idade encontra-se abaixo do limite permitido.

Vamos praticar?

Preencha os requisitos abaixo de acordo com um sistema de reserva de assentos em um cinema. Considere que a capacidade máxima é de 100 lugares.

Limite inferior

Limite superior

Fora do limite

Bloco 5

Hora de testar o que vimos até aqui!

Hora do exercício

Dividam-se em quatro grupos para realizar a atividade. Aqui, cada equipe deve receber um tema específico sobre os diferentes tipos de testes funcionais. Pesquisem as suas características na internet e preencham uma cartolina.

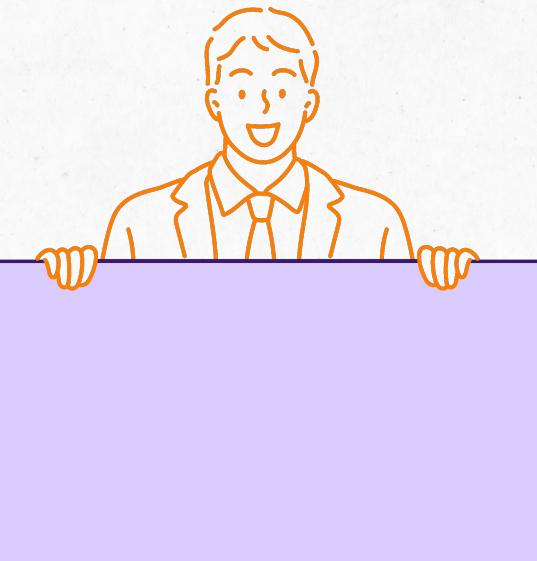
No bloco seguinte, vocês terão um momento para apresentar o que foi feito.

Unidade

Integração

Sistema

Aceitação



Preparem-se!

Chegou a hora de sortear!

Acesse o site para sortear os grupos de forma rápida e prática!



Bloco 6

Vamos apresentar!

Discussões e reflexões



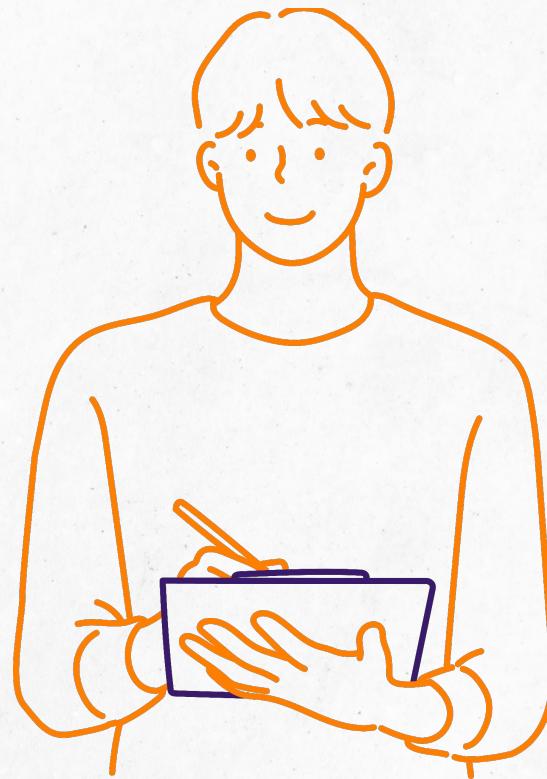
A large orange illustration occupies the left side of the slide. It depicts a teacher with long brown hair, wearing a white t-shirt and orange pants, standing and holding a megaphone to their mouth. A large orange speech bubble originates from the megaphone. Inside the speech bubble, a student with short brown hair, wearing a white t-shirt, is shown with their right hand raised towards the teacher's head. The teacher has a surprised or engaged expression, indicated by three small orange wavy lines above their head. The background behind the teacher is a light grey.

Hora dos *feedbacks*!

O(A) professor(a) irá avaliar a compreensão dos alunos a partir das apresentações em grupo, da participação nas discussões e das respostas dadas às perguntas feitas durante a atividade.

Fechamento

Se a aula de hoje passasse por um teste funcional, como ela seria avaliada?



Referências Bibliográficas

PROZ EDUCAÇÃO. *Apostila de Qualidade de software*. 2023.