

TESTE E QUALIDADE DE SOFTWARE

João Choma Neto

joao.choma@unicesumar.edu.br

ARTEFATOS

Os artefatos de teste são documentos e entregas produzidos durante o processo de teste de software

Esses artefatos têm o propósito de fornecer informações sobre o planejamento, execução e resultados dos testes

ARTEFATOS DE TESTE

<https://github.com/JoaoChoma/testedesoftware2025/tree/main/SEMANA01/docs>

PLANO DE TESTE

É um documento que descreve a abordagem geral de teste:

1. Objetivos do teste
2. Escopo
3. Cronograma
4. Recursos necessários e
5. Critérios de aceitação

CASOS DE TESTE

São documentos que descrevem os cenários específicos que serão testados

1. Entrada de dados
2. Ações a serem realizadas
3. Resultados esperados

ROTEIRO DE TESTE

São documentos que detalham a sequência de etapas que os testadores devem seguir ao executar os casos de teste

RELATÓRIO DE EXECUÇÃO DE TESTES

São documentos que resumem os resultados dos testes executados

1. Mostram quais casos de teste foram bem-sucedidos
2. Mostram quais casos de teste falharam
3. Medem o progresso do teste
4. Identificam quais áreas que exigem atenção do testador

MATRIZ DE RASTREABILIDADE

É uma tabela que mostra a relação entre os requisitos do software e os casos de teste

A matriz tem o objetivo de garantir que todos os requisitos foram cobertos por testes

CASO DE TESTE

- Um **caso de teste** é um documento que descreve de forma clara e objetiva uma **situação específica** que será testada em um sistema ou software, com o objetivo de **verificar se uma funcionalidade está funcionando conforme o esperado**.

CASO DE TESTE

- **Componentes de um Caso de Teste**

- Geralmente, um caso de teste contém os seguintes elementos:
- **Identificador (ID):** um código único para rastrear o caso (ex: CT-01)
- **Título ou Nome:** breve descrição do que está sendo testado (ex: “Agendar consulta com dados válidos”)
- **Pré-condições:** o que deve estar pronto antes do teste (ex: paciente já cadastrado)

CASO DE TESTE

- **Componentes de um Caso de Teste**

- **Entradas:** dados que serão inseridos (ex: data: 01/08/2025, hora: 14h, médico: Dr. João)
- **Passos de execução:** ações que o testador deverá realizar (ex: clicar em “Agendar”)
- **Resultado esperado:** o que o sistema deve fazer (ex: mostrar mensagem “Consulta agendada com sucesso”)
- **Resultado real** (preenchido na execução): o que aconteceu de fato
- **Status:** aprovado (passou) ou reprovado (falhou)

CLASSES DE EQUIVALENCIA

- **O que é uma Classe de Equivalência?**
- **É um conjunto de valores de entrada que o sistema trata da mesma forma, ou seja, se um valor da classe passar ou falhar, presume-se que todos os outros valores da mesma classe terão o mesmo comportamento.**

CLASSE DE EQUIVALENCIA

- Porque é **inviável testar todos os valores possíveis** de entrada. Em vez disso, testamos **um valor representativo de cada classe válida e inválida**.

CLASSE DE EQUIVALENCIA

lasse de Equivalência	Tipo	Valores Exemplares
Menores que 18	Inválida	10, 0, -5
Entre 18 e 65 (inclusive)	Válida	18, 30, 65
Maiores que 65	Inválida	70, 80, 100

CLASSE DE EQUIVALENCIA

- **Entradas com classes de equivalência:**
- Se a funcionalidade exige “idade entre 18 e 65”,
- Você testaria:
 - 1 valor da classe válida: idade = 30
 - 1 valor da classe inválida (abaixo): idade = 10
 - 1 valor da classe inválida (acima): idade = 70
 -
- Isso é mais **eficiente** do que testar todas as idades de 1 a 100.

REFERÊNCIAS

Ian Sommerville – Engenharia de Software. 10ª Edição. São Paulo: Pearson Education do Brasil, 2019.

Roger S. Pressman – Engenharia de software: uma abordagem profissional. 7ª Edição. Porto Alegre: AMGH Editora Ltda, 2011.

Shari Lawrence Pfleeger – Engenharia de Software: teoria e prática. 2ª Edição. São Paulo: Pearson Education do Brasil, 2004.