

## Template para Estudo de Caso de Testes Funcionais

**Conteúdo:** Processo de Engenharia de Software, Processo e Etapas da Atividade de Teste, Teste funcional / Teste de caixa preta, Plano de Teste e Projeto de Casos de Teste.

**Facilitador:** Docente nas disciplinas de Engenharia de Software e áreas afins

**Objetivos de Aprendizagem:** Deseja-se que ao final desse estudo o aluno seja capaz de:

- Compreender o processo de engenharia de software e a sua importância na construção de um produto de software sustentável;
- Compreender as etapas da atividade de teste;
- Compreender os princípios fundamentais dos testes funcionais e seu papel no ciclo de desenvolvimento de software;
- Ser capaz de criar casos de **teste funcionais detalhados com base em requisitos** de software;
- Realizar testes funcionais de acordo com planos de teste e documentar os resultados de forma precisa;
- Criar e desenvolver histórias para seus projetos em grupo;

**Estratégias de Uso:**

- É desejável que o facilitador faça uma breve apresentação sobre o processo de engenharia de software (Figuras 1 e 2);

- É desejável que o facilitador faça uma breve apresentação sobre as etapas da atividade de teste (Figuras 3 e 4);
- Apresentar o modelo de estudo de caso Estudo de Caso: Funcional - Valida Visibilidade de Campos disponível em :  
<https://docs.google.com/document/d/1N65molLxSZHBn3qbRDj6OBrOuvGQgZKyD3j0JEb32rE/edit?usp=sharing>
- Reunir os alunos em grupos (por projeto), e sugerir que eles selecionem dois ou mais requisitos de seus projetos, e, se possível, que já tenham desenvolvidos para planejamento e criação de testes funcionais;
- Sugerir que os alunos realizem o particionamento de equivalência do requisito escolhido;
- Sugerir aos alunos o desenvolvimento do Planejamento e o Projeto de Casos de Teste do requisito escolhido.

**Requisitos mínimos:** Conhecimento em lógica de programação e alguma linguagem de programação.



## GUIA

1. Processo de Engenharia de Software	4
2. Processo e Etapas da Atividade de Teste	6
3. Teste Funcional ou Caixa Preta: Definição	8
4. Planejamento/Plano de Teste e Projeto de Casos de Teste	10
5. Sugestões	12
6. Referências	13

# 1. Processo de Engenharia de Software

## 2.1 Um modelo de processo genérico

No Capítulo 1, *processo* foi definido como um conjunto de atividades de trabalho, ações e tarefas realizadas quando algum artefato de *software* deve ser criado. Cada uma dessas atividades, ações e tarefas se aloca dentro de uma metodologia ou modelo que determina sua relação com o processo e uma com a outra.

O processo de *software* está representado esquematicamente na Figura 2.1. De acordo com a figura, cada atividade metodológica é composta por um conjunto de ações de engenharia de *software*. Cada ação é definida por um *conjunto de tarefas*, o qual identifica as tarefas de trabalho que devem ser completadas, os artefatos de *software* que serão produzidos, os fatores de garantia da qualidade que serão exigidos e os marcos utilizados para indicar progresso.

### Processo de software

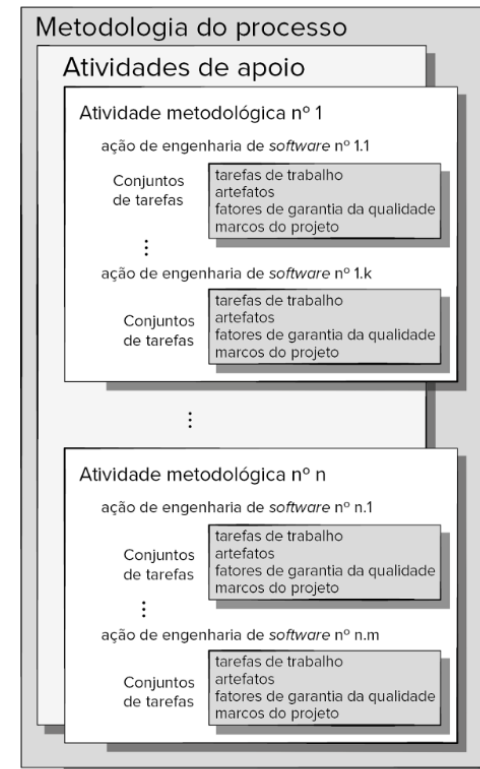


Figura 1: Figura 2.1 Engenharia de Software - 9ª Edição (PRESSMAN et al., 2021)

**Figura 2.1**

Uma metodologia do processo de *software*.

Como discutido no Capítulo 1, uma metodologia de processo genérica para engenharia de *software* estabelece cinco atividades metodológicas: **comunicação, planejamento, modelagem, construção e entrega**. Além disso, um conjunto de atividades de apoio é aplicado ao longo do processo, como o **acompanhamento e o controle do projeto, a administração de riscos, a garantia da qualidade, o gerenciamento das configurações, as revisões técnicas**, entre outras.

Um aspecto importante do processo de *software* ainda não foi discutido. Este aspecto – chamado de *fluxo de processo* – descreve como são organizadas as atividades metodológicas, bem como as ações e tarefas que ocorrem dentro de cada atividade em relação à sequência e ao tempo, como ilustrado na Figura 2.2.

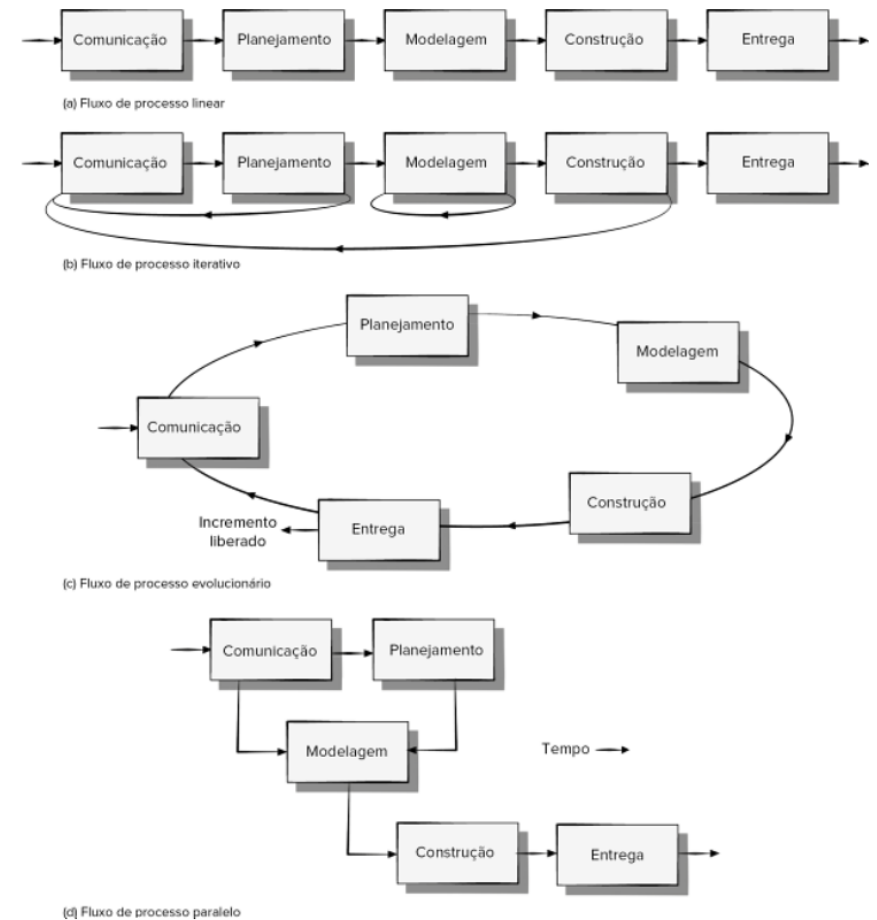


Figura 2: Figura 2.2 Engenharia de Software - 9ª Edição (PRESSMAN et al., 2021)

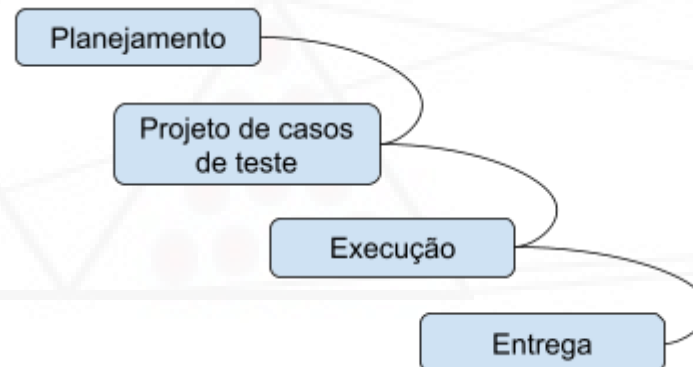
## 2. Processo e Etapas da Atividade de Teste

De modo geral, as etapas da atividade de teste são: planejamento/plano de teste, projeto de caso de teste, execução e análise.



Figura 3: Figura 8.3 Modelo do processo de teste de software 10ª Edição (SOMMERVILLE, 2019)

### Resumo das etapas da atividade de teste



Uma boa prática ao desenvolver um bom conjunto de testes, ou seja, o que irá revelar o maior número de defeitos, é construí-los paralelamente com as etapas do desenvolvimento como sugere a figura abaixo que traz o Modelo V, que faz parte dos modelos de Ciclo de Desenvolvimento de Software (SDLC, do inglês *software development life cycle*), que traz um exemplo prático coerente ao processo processo de construção do software.

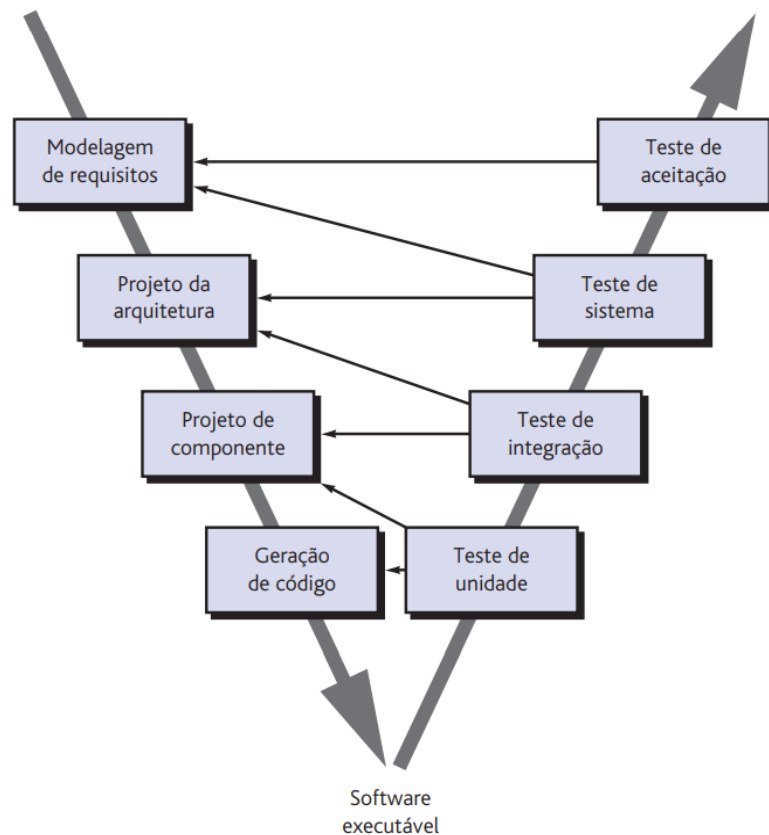


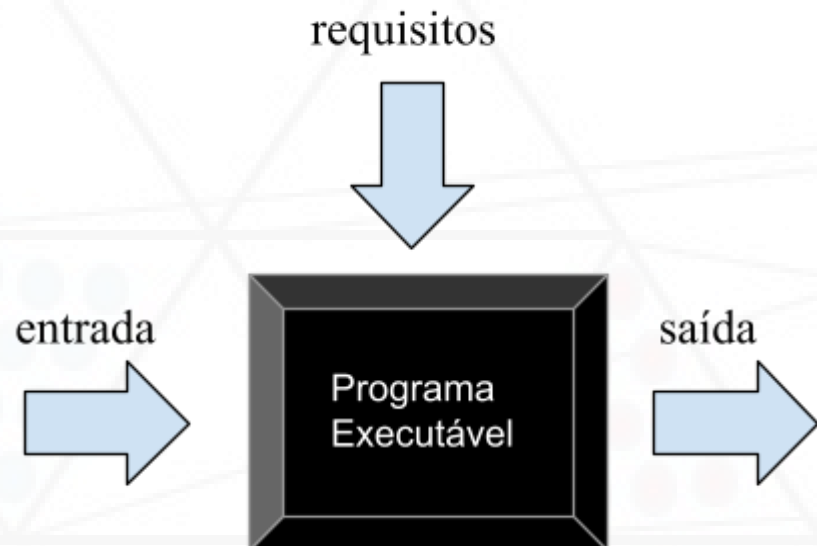
Figura 4: Figura 4.2. Modelo V Engenharia de Software - 8ª Edição (PRESSMAN; MAXIM, 2016)

### 3. Teste Funcional ou Caixa Preta: Definição

O teste funcional ou caixa preta é uma técnica de teste que visa identificar uma determinada se o comportamento do software ocorre como **especificado** para determinada funcionalidade implementada.

Dessa forma, é fornecido ao responsável pela realização do teste, um documento da especificação do requisito a ser testado não sendo necessário que o testador conheça a estrutura interna do código e sim tenha bem definida a regra de negócio do requisito.

A ideia principal desse tipo de teste é projetar casos de teste para garantir que o requisito funcional seja satisfatório.



- ✓ Foco no comportamento do requisito como esperado;
- ✓ O testador não precisa conhecer a estrutura interna do sistema para realizar o teste;

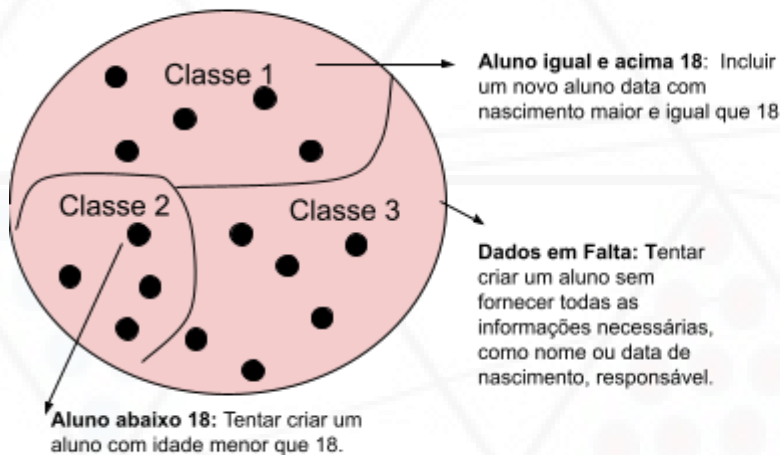


Um dos critérios de teste para projetar casos de teste funcionais é o Particionamento em Classe de Equivalência.

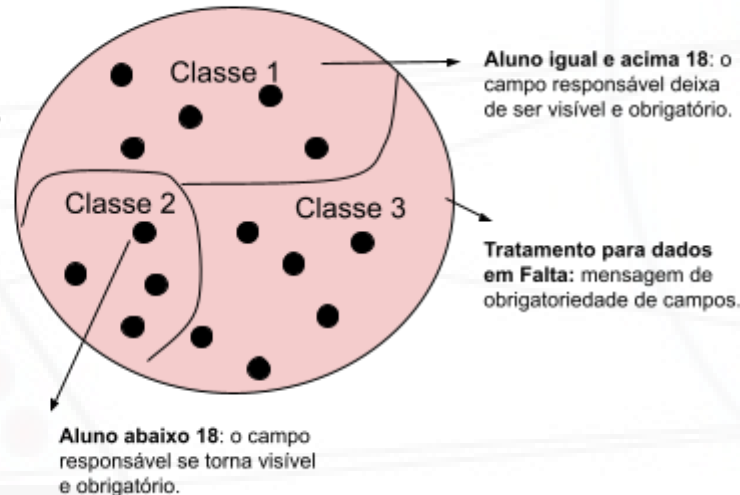
A classe de equivalência é um conceito comumente usado em testes de software e engenharia de qualidade. Ela é usada para agrupar conjuntos de entradas de teste que têm características semelhantes e que podem ser tratadas da mesma maneira durante os testes. O objetivo das classes de equivalência é garantir que você teste um conjunto representativo de valores de entrada, sem testar cada valor individualmente.

A ideia é dividir o espaço de entrada do sistema em grupos de valores que se comportam de maneira semelhante e, em seguida, testar apenas um valor de cada grupo. Se um valor de teste em uma classe de equivalência passar no teste, você pode razoavelmente supor que outros valores nessa mesma classe também serão bem-sucedidos.

Domínio de entrada - Visualização



Domínio de saída - Visualização



Como no teste funcional o foco é o comportamento do sistema com base no requisito, para iniciar o planejamento é necessário definir qual requisito do sistema deverá ser testado. Segue abaixo um modelo de especificação, onde foi selecionado o requisito RF1 para ser testado.

Especificação de Requisitos - Secretaria Escolar			
Requisito	Nome	Descrição	Status
RF1	Tela de Cadastro de Alunos	<ul style="list-style-type: none"><li>• Quando a idade do aluno for maior ou igual a 18 anos, o campo responsável <b>não será visível e deixará de ser obrigatório.</b></li><li>• Quando a idade do aluno for menor a 18 anos, o campo responsável <b>será visível e obrigatório.</b></li></ul>	Em homologação
...			

#### 4. Planejamento/Plano de Teste e Projeto de Casos de Teste

O planejamento ou plano de teste e o projeto de casos de teste são fatores que propõem uma melhor rastreabilidade e segurança de todo o processo de teste.

O IEEE (Instituto de Engenheiros Eletricistas e Eletrônicos), sugere um padrão (“ISO/IEC/IEEE 29119-3:2021(en), Software and systems engineering — Software testing — Part 3: Test documentation”, [s.d.]) de planejamento e especificação dos casos de teste que pode ser adaptado para cada organização ou projeto.

As tabelas abaixo são uma simplificação do modelo sugerido pela IEEE.

### Planejamento - Secretaria Escolar

O que testar?	Como testar?	Qual a base de dados?	Quem irá testar?	Cronograma	Outros(riscos, requisitos do ambiente)
Ex.: RF1 Tela de Cadastro de Alunos	Qual critério de teste? 💡 <b>Sugestão:</b> <ul style="list-style-type: none"> <li>- particionamento de equivalência;</li> <li>- e/ou análise de valores limites;</li> </ul>	Conjunto de dados que serão utilizados como entrada e formato  Ex.: csv, json, txt, docx, xls, etc.	Usuário com perfil de secretaria e atendimento.	Dia XX/XX/2023 a Dia XX/XX/2023	Riscos: Não de Aplica (N/A). Requisitos do Ambiente: <ul style="list-style-type: none"> <li>- Python - Extensão Selenium IDE</li> </ul>

### Projeto de caso de teste - Secretaria Escolar

ID	Caso de Teste	Descrição	Roteiro	Comportamento esperado	Comportamento real, evidências e incidentes.	Responsável	Status
----	---------------	-----------	---------	------------------------	--	-------------	--------

	Classe 1	Aluno maior ou igual a 18.	Aqui deverá ser preenchidas as <b>entradas</b> das classes de equivalência	Aqui deverá ser preenchidas as <b>saídas</b> das classes de equivalência	Concluída com sucesso. [link imagem]	José	Concluído
	Classe 2	Aluno menor que 18.	Aqui deverá ser preenchidas as <b>entradas</b> das classes de equivalência	Aqui deverá ser preenchidas as <b>saídas</b> das classes de equivalência	Falha: ao salvar com sem preencher o campo "nome". [link imagem]	Maria	Erro
	Classe 3	Dados em falta.	Aqui deverá ser preenchidas as <b>entradas</b> das classes de equivalência	Aqui deverá ser preenchidas as <b>saídas</b> das classes de equivalência	Falha: Não está sendo possível ver o campo responsável quando o aluno é menor de 18. [link imagem]	José	Erro
	....						

## 5. Sugestões

💡 Sugestão de Ferramentas para Teste Funcionais.

- Java, Python, Ruby, JS - Appium: <https://appium.io/docs/en/2.1/>
- Python - Selenium : <https://selenium-python.readthedocs.io/>;
- Python - Extensão Selenium IDE : <https://www.selenium.dev/selenium-ide/>
- NodeJS - Cypress: <https://docs.cypress.io/guides/overview/why-cypress>

💡 Sites de Geração de Dados para Teste:

- [generatedata.com](https://generatedata.com)
- [Mockaroo - Random Data Generator and API Mocking Tool | JSON / CSV / SQL / Excel](https://Mockaroo.com)
- [Online test data generator for up to 100.000 Records \(onlinedatagenerator.com\)](https://onlinedatagenerator.com)
- [Cobbl: Mock Data Generator](https://Cobbl.com)
- [Random Mock Data Generator \(randomtools.io\)](https://randomtools.io)

## 6. Referências

**Ejemplos de relaciones de equivalencia no triviales, me refiero a relaciones de equivalencia sin la expresión “igual ... que” en su definición.** Disponível em:

<<https://isolution.pro/es/q/ma16202124/ejemplos-de-relaciones-de-equivalencia-no-triviales-me-refiero-a-relaciones-de-equivalencia-sin-la-expresion-igual-q>>. Acesso em: 5 nov. 2023.

PRESSMAN, R.; MAXIM, B. **Engenharia de Software - 8ª Edição**. [s.l: s.n.].

PRESSMAN, R. S. et al. **Engenharia de software**. 9ª edição ed. Porto Alegre, RS: AMGH, 2021.

**Snapshot.** , [s.d.]. Disponível em:

<<https://isolution.pro/es/q/ma16202124/ejemplos-de-relaciones-de-equivalencia-no-triviales-me-refiero-a-relaciones-de-equivalencia-sin-la-expresion-igual-q>>. Acesso em: 5 nov. 2023

SOMMERVILLE, I. **Engenharia de Software**. 10ª edição ed. [s.l.] Pearson Universidades, 2019.