

# Introdução ao Teste de Software

## Apresentação da Disciplina

Vinicius H. S. Durelli

✉ [durelli@ufsj.edu.br](mailto:durelli@ufsj.edu.br)



# Organização

- 1 Objetivos da disciplina
- 2 Pré-requisitos
  - “Regras do jogo”
  - Metodologia, recursos e livro-texto
- 3 Avaliação e cronograma
  - Exercícios em sala
  - Trabalho Prático
- 4 Considerações finais

## 1 Objetivos da disciplina

## 2 Pré-requisitos

- “Regras do jogo”
- Metodologia, recursos e livro-texto

## 3 Avaliação e cronograma

- Exercícios em sala
- Trabalho Prático

## 4 Considerações finais

## Objetivos da disciplina (1)

Introduzir os alunos aos **conceitos fundamentais** relacionados ao teste de software.

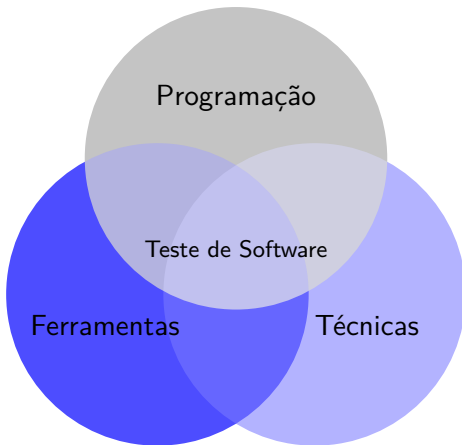
- **Importância** da atividade no cenário atual.
- Noções sobre **técnicas e critérios de teste**.
- **Técnicas de desenvolvimento** (projeto/análise) que utilizam testes: *Test-driven development* (TDD) e *Behavior-driven development* (BDD).
- **Ferramentas** que são utilizadas para automatização de atividades de teste de software.
  - Criação de testes.
  - *Mock objects* (objetos *mock*).

**Objetivo:** ensinar o aluno a **pensar e atuar como um testador**.

- **Teoria/Filosofia:** entender teste de software (*Beizer's Testing Levels*).
- **Teoria:** ter conhecimento das diferentes técnicas e critérios que podem ser aplicados para guiar a atividade de teste de software.
- **Prática/automação:** utilizar ferramentas para automatizar (controlar a execução) de testes facilitar a condução dessa atividade.

# Objetivos da disciplina (2)

**Objetivo:** ensinar o aluno a **pensar e atuar como um testador**.



## Pré-requisitos. . .

Conforme ferramentas e critérios de teste forem apresentados, algumas vezes, conceitos fundamentais não serão recapitulados. Portanto, assume-se conhecimento prévio em:

- Lógica;
- Matemática discreta;
- Engenharia de software;
- Programação orientada a objetos.



## Algumas observações. . .

*Pensamento do dia:* a disciplina não é obrigatória, portanto, ela é menos importante, menos complicada e demanda menos tempo. Certo?!

**Pelo contrário:** a indústria tem despertado para a extrema importância da atividade de teste e, portanto, vejam isso como uma oportunidade.

- É diferente de tudo que vocês já aprenderam.



## Algumas observações. . .

*Pensamento do dia:* a disciplina não é obrigatória, portanto, ela é menos importante, menos complicada e demanda menos tempo. Certo?!

**Pelo contrário:** a indústria tem despertado para a extrema importância da atividade de teste e, portanto, vejam isso como uma oportunidade.

- **É diferente de tudo que vocês já aprenderam.**
- **Indispensável** no processo de desenvolvimento de software.

## Algumas observações. . .

*Pensamento do dia:* a disciplina não é obrigatória, portanto, ela é menos importante, menos complicada e demanda menos tempo. Certo?!

**Pelo contrário:** a indústria tem despertado para a extrema importância da atividade de teste e, portanto, vejam isso como uma oportunidade.

- **É diferente de tudo que vocês já aprenderam.**
- **Indispensável** no processo de desenvolvimento de software.
- **Escassez de mão-de-obra** especializada na área.

## Algumas observações. . .

*Pensamento do dia:* a disciplina não é obrigatória, portanto, ela é menos importante, menos complicada e demanda menos tempo. Certo?!

**Pelo contrário:** a indústria tem despertado para a extrema importância da atividade de teste e, portanto, vejam isso como uma oportunidade.

- **É diferente de tudo que vocês já aprenderam.**
- **Indispensável** no processo de desenvolvimento de software.
- **Escassez de mão-de-obra** especializada na área.
- **Área de pesquisa** que vem recebendo bastante atenção.

# As regras do jogo...

Durante as aulas:

- **Tentar** ser pontual.
- Desligar o celular!
- Evitar o uso de linguagem imprópria/vulgar.
- Respeitar as ideias/comentários dos outros.

Em casa:

- **NÃO** copiar exercícios alheios.
- Fazer os exercícios!<sup>a</sup>

---

<sup>a</sup>Faça os exercícios ao invés de somente revisar o material.



É importante enfatizar que...

# Dedicação é fundamental!



Toda<sup>1</sup> aula envolve a resolução de pelo menos um exercício.

---

<sup>1</sup>Bom, quase toda aula.

## Em caso de dúvidas. . .

Pessoalmente:

- Sala 3.05 (se possível, **agendar antes**).
- Horários de atendimento:
  - Terças 16h-17h;
  - Quintas 15h-17h;

Para as pessoas introvertidas:

- `vinicius.durelli@gmail.com`; ou
- `durelli@ufsj.edu.br`.



# Metodologia e recursos

## Aulas

- Aulas teóricas expositivas.<sup>a</sup>

<sup>a</sup> Algumas aulas conciliam teoria e prática de maneira dinâmica.

## Recursos

- Lousa, projetor multimídia e computador.

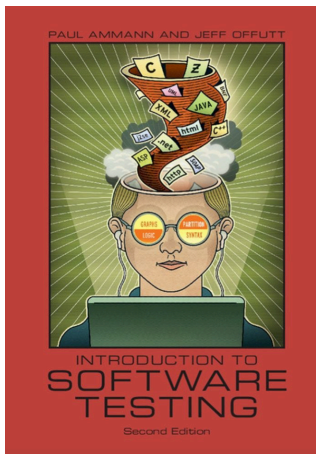
## Prática

- Exercícios e trabalhos práticos (TPs).

# Livro-texto 1: Introduction to Software Testing

Paul Ammann e Jeff Offutt

Para uma discussão completa dos vários critérios de teste discutidos no decorrer da disciplina (Ammann & Offutt 2008; Ammann & Offutt 2016):



Paul Ammann & Jeff Offutt (2008). *Introduction to Software Testing*. 1st ed. Cambridge University Press, p. 346

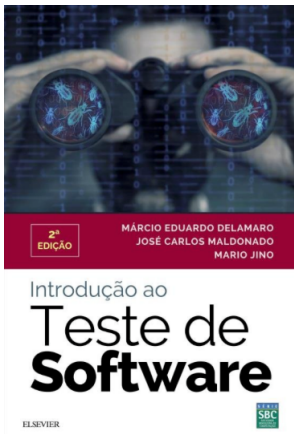
Paul Ammann & Jeff Offutt (2016). *Introduction to Software Testing*. 2nd ed. Cambridge University Press, p. 364



# Livro-texto 2: Introdução ao Teste de Software

Márcio Delamaro, Mario Jino e José Carlos Maldonado

Para uma discussão tanto teórica quanto técnica dos assuntos abordados no decorrer da disciplina (Delamaro et al. 2016):



Marcio Delamaro et al. (2016).  
*Introdução ao Teste de Software*. 2nd ed. Elsevier, p. 394

- 1 Objetivos da disciplina
- 2 Pré-requisitos
  - “Regras do jogo”
  - Metodologia, recursos e livro-texto
- 3 **Avaliação e cronograma**
  - Exercícios em sala
  - Trabalho Prático
- 4 Considerações finais

- **Exercícios em sala e no laboratório** ( $E_1, E_2, \dots, E_n$ ) valendo 50% da nota final (NF).<sup>a</sup>
- **Um TP** valendo 50% da NF.<sup>b</sup>

$$NF = \sum_{i=1}^n \frac{E_i}{n} \times 5 + TP \times 5$$

---

<sup>a</sup>Exercícios teóricos valem 20% e os em laboratório valem 30%.

<sup>b</sup>Primeira parte vale 20% e a segunda 30%.

**Aprovado:** aluno com  $NF \geq 6$ .

**Prova substitutiva:** (somente) para o aluno com  $NF < 6$   
(essa prova **substitui somente** a nota do **TP**).

## Um pouco mais sobre os exercícios em sala (1)...

Conforme ferramentas e critérios de teste forem apresentados, teremos exercícios em sala (ou laboratório):

- Geralmente, devem **começar e terminar** em sala;
- *Feedback* e nota na aula (ou semana) seguinte;
- Avaliadas da seguinte forma:
  - 0 → não entregou;
  - 2–4 → completada em menos de 50%;
  - 5–9 → completada de 51% até 99%;
  - 10 → completada em 100%;



## Um pouco mais sobre os exercícios em sala (2)...

Todos exercícios em sala são individuais?

- Exercícios em sala não identificados como individual podem ser feitos em grupos (de no máximo 3 alunos).
  - Colocar no documento ou e-mail: *“Fulano com o parça Ciclano”* ou ainda *“Fulanete com a bestie Ciclanete”*.
- Lembrete: **ética acadêmica**.

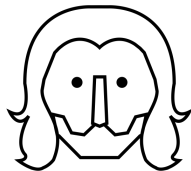
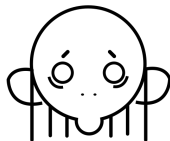
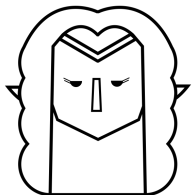


## Um pouco mais sobre o trabalho prático (1)...

Conforme mencionado, o trabalho prático corresponde à 50% da nota da disciplina.

- Será realizado em grupos de 4 alunos;
- “Entregas”/apresentações parciais serão solicitadas ao longo do semestre;
- Datas relacionadas ao TP aparecem na programação da disciplina.
- Tentar alinhar o TP com interesses pessoais. disciplina.

## Escolham bem os membros do grupo! (The Fellowship of the Ring TP)



E, por fim, o cronograma proposto. . .

# Cronograma: Teste de Software



## 1 Objetivos da disciplina

## 2 Pré-requisitos

- “Regras do jogo”
- Metodologia, recursos e livro-texto

## 3 Avaliação e cronograma

- Exercícios em sala
- Trabalho Prático

## 4 Considerações finais

## Considerações finais. . .

Na aula de hoje nós vimos:

- Objetivo(s) da disciplina;
- Relevância desta disciplina;

Na **próxima aula**: conceitos básicos de teste de software.

**(Lista com os grupos)**

Ammann, Paul & Jeff Offutt (2008). *Introduction to Software Testing*. 1st ed. Cambridge University Press, p. 346.

– (2016). *Introduction to Software Testing*. 2nd ed. Cambridge University Press, p. 364.

Delamaro, Marcio, Mario Jino, & Jose Carlos Maldonado (2016). *Introdução ao Teste de Software*. 2nd ed. Elsevier, p. 394.

"TODO List" icon by Isabel Foo, "Studying Desk" icon by Delwar Hossain, "Abacus" icon by Artem Kovyazin, "Working (alone)" icon by Adrien Coquet, "Working (in collaboration)" icon by Björn Andersson, "Elf", "Warrior", "Gollum", and "Hobbit" icon by Rodrigo Vidinich, from the Noun Project (<https://thenounproject.com/>).