

MC646 – Verificação, Validação e Testes

(1º semestre 2022)

Prof. Breno Bernard Nicolau de França

breno@ic.unicamp.br

www.ic.unicamp.br/~breno

Dia	Horário
Segunda-feira	19h-21h (353)
Quarta-feira	21h-23h (353)
Atendimento¹ (com horário marcado)	Segundas 14h-16h

1. Objetivos Terminais

Ao final do curso, o aluno deve ser capaz de:

- ★ Analisar os requisitos de um sistema de software e decidir sobre as técnicas oportunas para verificá-los e validá-los.
- ★ Aplicar corretamente técnicas de verificação e validação de software.

2. Conhecimentos Requeridos

- Programação modular em linguagem com suporte a definição de interfaces e componentes (módulos), ou linguagem com suporte a orientação a objetos;
- Análise e Projeto de Software, que compreende a engenharia de requisitos, bem como as derivações de requisitos em artefatos de arquitetura/projeto e código.

3. Unidades

3.1. Conceitos de Verificação e Validação

Objetivo	O aluno deve ser capaz de compreender os conceitos de qualidade de software, verificação e validação, e suas implicações no ciclo de vida do software.
Procedimento	Aula expositiva dialogada ² e leituras de capítulo ³ do livro.

¹ Outros horários podem ser agendados com o professor mediante disponibilidade.

² Este modelo de aula expande o conceito da aula expositiva tradicional, trazendo o aluno para uma discussão do conteúdo exposto, podendo incitar questionamentos, análise crítica e confronto com a realidade. Práticas: convidar alunos a responder questões, formar grupos para discutir e apresentar uma visão sobre um tópico; pedir para alunos levantarem questões; discussão de práticas de como o objeto aparece em cenários reais.

³ Capítulo 24. Ian Sommerville. Software Engineering, 9 Ed. 2011. Pearson.

<i>Avaliação</i>	Lista de questões discursivas. Discussão de possíveis respostas no gabarito. Entrega individual.
------------------	--

3.2. Técnicas de Verificação Estática

<i>Objetivo</i>	O aluno deve ser capaz de analisar os diferentes tipos de técnicas de verificação estática: revisão, inspeção e análise estática de código.
<i>Procedimento</i>	Aula expositiva dialogada e duas leituras (inspeções e análise estática).
<i>Avaliação</i>	Análise comparativa de revisões baseadas em checklists e análise estática. Entrega em dupla.

3.3. Fundamentos de Teste de Software

<i>Objetivo</i>	O aluno deve ser capaz de compor uma estratégia de testes a partir de um contexto de projeto, utilizando os conceitos fundamentais de teste de software, dimensões, bem como o processo de testes.
<i>Procedimento</i>	Aula expositiva dialogada, lista de exercícios.
<i>Avaliação</i>	Definir uma estratégia de testes a partir de um cenário hipotético. Mínimo duas páginas explicando a estratégia e justificando as decisões tomadas. Entrega em dupla.

3.4. Test-Driven e Behavior-Driven Development

<i>Objetivo</i>	O aluno deve ser capaz de aplicar as práticas de TDD e BDD no desenvolvimento de software.
<i>Procedimento</i>	Aula expositiva dialogada, laboratórios/exercícios práticos.
<i>Avaliação</i>	Desenvolver uma função especificada e os casos de teste para a mesma utilizando a técnica TDD em laboratório. Entrega dos casos de teste e a função codificados (com comentários) A presença do aluno nesta avaliação é obrigatória.

3.5. Técnicas de Teste de Software

<i>Objetivo</i>	O aluno deve ser capaz de aplicar as técnicas de testes de software para o projeto de casos de testes.
<i>Procedimento</i>	Aula expositiva dialogada, laboratórios/exercícios práticos.

Avaliação	<p>Adicionar casos de testes relevantes para um software livre existente com base em pelo menos três das seguintes técnicas: Análise de Valor Limite, Particionamento em Classes de Equivalência, Tabela de Decisão, Grafos de Causa e Efeito, Pairwise, Análise de Fluxo de Dados, e Análise de Fluxo de Controle.</p> <p>Entrega dos casos de teste codificados e um breve relatório (documento PDF) explicando como os mesmos foram derivados/gerados.</p>
------------------	---

3.6. Testes Não-Funcionais

Objetivo	O aluno deve ser capaz de aplicar testes não funcionais para as características: Desempenho (Estresse e Carga), Robustez (Injeção de Falhas), e Segurança.
Procedimento	Aula expositiva dialogada, laboratórios/exercícios práticos.
Avaliação	<p>Implementar uma pequena aplicação Web ou Mobile (ou evoluir uma existente) e adicionar casos de testes relevantes com base nas seguintes características: desempenho, robustez, e segurança.</p> <p>Entrega dos casos de teste codificados e um breve relatório (documento PDF) explicando como os mesmos foram derivados/gerados.</p>

3.7. Teste de Mutação

Objetivo	O aluno deve ser capaz de aplicar técnicas de teste de mutação.
Procedimento	Aula expositiva dialogada, laboratórios/exercícios práticos.
Avaliação	<p>Derivar um conjunto de casos de testes utilizando técnicas de teste de mutação.</p> <p>Entrega dos casos de teste codificados e um relatório (documento PDF) explicando como os mesmos foram derivados/gerados.</p>

3.8. Teste Dirigido por Modelos

Objetivo	O aluno deve ser capaz de aplicar técnicas de teste dirigidos por modelos para diagramas de estados e especificações em casos de uso.
Procedimento	Aula expositiva dialogada, laboratórios/exercícios práticos.
Avaliação	Derivar um conjunto de casos de testes a partir de modelos de transição de estados e/ou casos de uso.

	Entrega dos casos de teste codificados e um relatório (documento PDF) explicando como os mesmos foram derivados/gerados.
--	--

4. Critérios de Avaliação

A avaliação da disciplina realizada com base nas avaliações das unidades: as notas serão atribuídas com base no desempenho do aluno para as atividades de avaliação previstas para cada unidade (seção 3). Todas as outras unidades terão peso 1 (10% da nota final), exceto pelas avaliações das unidades 2 e 5, que terão peso 2. As datas das avaliações e prazos de entrega estão definidos no cronograma (seção 5).

A média (M) da disciplina será calculada como:

$$M = 0.1U1 + 0.2U2 + 0.1U3 + 0.1U4 + 0.2U5 + 0.1U6 + 0.1U7 + 0.1U8$$

Exame:

- Caso no final da disciplina o aluno tenha média $2.5 \leq M < 5.0$, poderá fazer o exame (E), cuja data está no cronograma.
- Neste caso, a média final será calculada como $MF = (M+E)/2$, com nota máxima igual a 5.0.

4.1. Informações Importantes:

- As datas referentes às entregas, tanto dos projetos quanto dos exercícios, estão disponíveis no cronograma da disciplina.
- A presença é **obrigatória** em todas as aulas (incluindo laboratório). Frequência inferior a 75% causa reprovação.
- Casos de plágio (cópia de texto, imagem ou ideia) entre os trabalhos ou de conteúdos externos serão tratados com rigor. A nota da avaliação em questão será anulada sem possibilidade de reposição e o caso será encaminhado à coordenação do curso.

5. Cronograma

O cronograma conta com mais de 30 datas previstas, de forma que algumas podem ser utilizadas em caso de imprevistos no semestre (identificadas por “Reserva”). As datas definidas a seguir podem sofrer alterações devido a imprevistos e/ou situações adversas.

Data	Tópico
14/03	Apresentação da Disciplina
16/03	Conceitos de Verificação e Validação
21/03	Conceitos de Verificação e Validação

23/03	Técnicas de Verificação Estática: Revisão, Inspeção e Análise Estática de Software
28/03	Entrega Avaliação da Unidade 1
	Técnicas de Verificação Estática (Lab)
30/03	Técnicas de Verificação Estática (Lab)
04/04	Fundamentos de Teste de Software
06/04	Fundamentos de Teste de Software
11/04	Entrega Avaliação da Unidade 2
	TDD
13/04	TDD (Lab)
18/04	Entrega Avaliação da Unidade 3
	TDD (Lab)
20/04	BDD
25/04	BDD (Lab)
27/04	BDD (Lab)
02/05	Técnicas de Teste Funcional
04/05	Avaliação da Unidade 4 (Em Sala)
09/05	Técnicas de Teste Funcional (Lab)
11/05	Técnicas de Teste Funcional (Lab)
16/05	Técnicas de Teste Funcional (Lab)
18/05	Técnicas de Teste Estrutural
23/05	Técnicas de Teste Estrutural (Lab)
25/05	Técnicas de Teste Estrutural (Lab)
30/05	Testes Não Funcionais: Desempenho
01/06	Testes Não Funcionais: Desempenho (Lab)
06/06	Testes Não Funcionais: Robustez
08/06	Entrega da Avaliação da Unidade 5
	Testes Não Funcionais: Segurança
13/06	Testes Não Funcionais: Segurança (Lab)

15/06	Teste de Mutação
20/06	Teste de Mutação (Lab)
22/06	Entrega da Avaliação da Unidade 6
	Teste Baseado em Modelos
27/06	Teste Baseado em Modelos (Lab)
29/06	Entrega da Avaliação da Unidade 7
	Teste Baseado em Modelos (Lab)
04/07	Reserva
06/07	Entrega da Avaliação da Unidade 8
11/07	Reserva
13/07	Reserva
25/07	Exame

6. Bibliografia

O curso é baseado nos seguintes livros texto, ou edições mais novas dos mesmos. Qualquer material adicional de leitura será anunciado, em sala, quando necessário.

- Sommerville, I. (2011). *Engenharia de Software*, 9ª edição. São Paulo: Pearson Addison-Wesley, 22, 103.
- Delamaro, M., Jino, M., & Maldonado, J. (2017). *Introdução ao teste de software*. Elsevier Brasil.
- Ammann, P., & Offutt, J. (2016). *Introduction to software testing*. Cambridge University Press.