

## Programa de Ensino

---

### 1) Identificação

**Disciplina:** INE5448 - Tópicos Especiais em Aplicações Tecnológicas I  
**Carga horária:** 72 horas-aula    Teóricas: 72    Práticas: 0  
**Período:** 1º semestre de 2017 até a presente data

### 2) Cursos

- Ciências da Computação (208)
- Engenharia, áreas Elétrica e Mecânica, habilitação Controle e Automação (220)

### 3) Requisitos

- Ciências da Computação (208)
  - INE5417 - Engenharia de Software I
  - INE5608 - Análise e Projeto de Sistemas

### 4) Ementa

Ementa livre para assuntos relevantes na área de Aplicações Tecnológicas.

### 5) Objetivos

**Geral:** Obter uma visão geral dos principais conceitos relacionados ao teste de software, incluindo técnicas e ferramentas.

**Específicos:**

- Conhecer os diferentes níveis de testes de software.
- Conhecer diferentes técnicas de testes de software.
- Conhecer diferentes processos de testes de software.
- Utilizar diferentes processos e técnicas em exercícios práticos envolvendo testes de software.

### 6) Conteúdo Programático

6.1) Conceitos Básicos sobre Testes de Software [4 horas-aula]

6.2) Níveis de Testes [8 horas-aula]

- Testes de Unidade
- Testes de Integração
- Testes de Sistema
- Testes de Aceitação

6.3) Tipos de Testes [12 horas-aula]

- Testes de Regressão
- Testes de Funcionalidade
- Testes de Performance
- Testes de Escalabilidade
- Teste de Performance
- Teste de Usabilidade
- Testes de Carga e Estabilidade
- Testes de Regressão
- Testes de Documentação

- 6.4) Técnicas e Critérios de Testes [16 horas-aula]
  - Inspeção e walkthrough
  - Teste de Fluxo de Controle
  - Teste de Fluxo de Dados
  - Teste de Domínio
  - Teste Exploratório
  - Teste de Estado
  - Testes Baseados em Modelos
  - Teste de Mutação
- 6.5) Planejamento e Execução de Testes de Software [4 horas-aula]
- 6.6) Desenvolvimento Orientado a Testes [12 horas-aula]
- 6.7) Automação de Testes [4 horas-aula]
- 6.8) Ferramentas de Teste de Software [12 horas-aula]

## 7) Bibliografia Básica

- Beck, Kent. Test-Driven Development: By Example. Addison-Wesley, 2002.
- Delamaro, Márcio Eduardo; Jino, Mario; Maldonado, José Carlo. Introdução ao Teste de Software. Rio de Janeiro: Elsevier, 2007. ISBN 978-85-352-2634-8
- Myers, Glenford J. The art of software testing. 2nd ed. Ed. John Wiley & Sons, 2004. 234 p. ISBN 0471469122.
- Kshirasagar, Naik; Priyadarshi, Tripathy. Software Testing and Quality Assurance – Theory and Practice. Ed. John Wiley & Sons, 2008. 616p.

## 8) Bibliografia Complementar

- Bastos, Anderson. Base de conhecimento em teste de software. Niterói: Traço & Photo, 2006. (Acervo: 681.31 B299)
- Beck, Kent; Andres, Cynthia. extreme Programming: Explained – Second Edition, Addison-Wesley, 2004.
- Crispin, Lisa. Agile testing: a practical guide for testers and agile teams. Ed Pearson Education, 2009. 533 p. ISBN: 9780321534460.a
- Fowler, M. Refactoring: Improving the Design of Existing Code. Addison-Wesley, 1999.
- Gärtner, M. ATDD By Example – A Practical Guide to Acceptance Test-Driven Development. Addison-Wesley Pearson Education, p. 212, 2013.
- Highsmith, Jim. Agile Software Development Ecosystems. Ed. Addison Wesley, 2002.
- Meszaros, G. xUnit Test Patterns: Refactoring Test Code. Addison-Wesley, 2007. 833 p. ISBN 978-0131495050.
- Pressman, Roger S. Engenharia de software: uma abordagem profissional. 7. ed. Porto Alegre: AMGH, 2011. 780 p. ISBN 9788563308337.
- Sommerville, Ian. Engenharia de software. 9. ed. São Paulo: Pearson; c2011. xii, 529 p. ISBN 9788579361081.
- Wazlawick, Raul Sidnei. Análise e design orientados a objetos para sistemas de informação: modelagem com UML, OCL e IFML. 3. ed. Rio de Janeiro: Elsevier: Campus, c2015. 462 p. ISBN 9788535279849.
- Wazlawick, Raul Sidnei. Engenharia de software: conceitos e práticas. Rio de Janeiro: Elsevier; Campus, c2013. xxii, 343 p. ISBN 9788535260847.
- Artigos.